April 27, 2021

Raman Spectroscopy using a Concatenated Laser System to Measure Water Content and Chemicals in Fuel

Department of Electrical Engineering and Computer Science

College of Optics and Photonics

University of Central Florida

Dr. Lei Wei, Dr. Samuel Richie, Dr. Aravinda Kar

Sponsors: Ocean Insight, Innovative Photonic Solutions

Group 1

Yaelle Olivier	*PSE*
Tyler Morris	*CPE*
Austin Collins	*CPE*

Table of Contents

1. Executive Summary

When you say fossil fuels in our society today it usually has a negative connotation attached. It is usually followed by global warming, pollution, and emissions. As technology is improving, our society is becoming more conscious and there is a push to be environmentally friendly. There are now more engines and vehicles that are moving away from traditional gasoline or diesel and more towards electric alternatives. Although this is a great first step, we are still a while away from eliminating the use of fossil fuels completely. It should also be noted that without fuel our world would not be as advanced as it is today. Fuel is still an extremely reliable source of power especially in natural disasters and for heavy machinery. With this said the number one reason for equipment failure in natural disaster relief programs is due to bad fuel. This is not because the fuel is bad or unreliable, but because this gasoline or diesel is from emergency storage tanks that are often not checked regularly or before use in a natural disaster situation. When storage tanks that contain fuel go unchecked, they can start to collect water, and eventually what water will develop bacteria that feed off the fossil fuels creating clusters called sludge. This water contamination is the first step in causing bacteria and other growths in the fuel to make it unusable.

In today's market, there are kits for detecting water contamination or bacterial growth in the fuel. However, these kits either use chemical reactions with the fuel and/or take a long time for the results. Many of these are based on a qualitative measurement such as color. We think there could be a more effective way to measure fuel quantitatively in a short amount of time without using chemicals. This is where we would use optical technology and machine learning to do so, by use of Raman Spectroscopy using a Concatenated Laser system (RSCL). This system can detect water content among substances and chemicals by using a Raman spectroscopy setup with two lasers rather than the typical one laser system. There are Raman spectroscopy systems already created to find chemicals in fuel, however, there is only proof of one current Raman probe that uses a dual-wavelength system by Innovative Photonic Solutions (IPS). They hold the patent for the probe.

Although it is usually not as common to have a senior design project that is still a very new technology and not much research is out about it, we are very lucky to have sponsors to help us move forward with this project idea. It would not have been possible to do this without our main sponsor, Ocean Insight, providing the spectrometer, other optical elements, and most importantly knowledge and advice. Another crucial sponsor is IPS providing us with the proper laser system for this exact purpose.

The RSCL will be portable and easy to use within minutes, for quick fuel analyzations on the spot in the field. It will be made with the user and safety in mind, whether it be a technician or an engineer. Looking further on the RSCL can be used for more purposes than just fuel. It could be accommodated to find water in different chemicals and substances.

2. Project Narrative

In this section, we will explore the deeper motivation for developing a Raman spectroscopy system to measure water content in fuel. We will go through the background of why it is so damaging to have water present in the fuel as well as the differences between using the “one and done” kits against a reusable quantitative way of measuring. Through this, we will also go through the engineering and marketing specifications to make sure that the final product is feasible and safe. To have a better understanding of our Raman Spectroscopy using a concatenated laser system (RSCL) there are block diagrams that walk through the optical diagram, hardware diagram, and software flowchart.

2.1 Project Background

Water contamination is the silent and often overlooked disease for engines using any petroleum fuel source. In an emergency using fuel reserves, the number one issue for equipment failure is due to the fuel being contaminated. This is because the government and fuel companies will buy and store large amounts of fuel and store it for a long time in case emergencies happen. However, the tanks often go unchecked and the fuel fails in such an emergency. Fuel just isn’t what it used to be; it has been modified through refineries to produce more fuel for less crude oil. This means that that the diesel you get now has a much shorter shelf life of 90 days compared to the shelf life in the 1960s in which the shelf life was 10 years. It has been shown that diesel will degrade 26% in the first month and will degrade to 95% in the same time frame if there is any water present in it. Another change to diesel fuel was the move towards ultra-low sulfur diesel (ULSD) to prevent sulfur pollution in the environment. Sulfur is a known anti-microbial, so today's diesel fuels are much more likely to easily be affected by microbes and water due to the minimizing of sulfur.

Although microbes in a fuel tank do not sound very dangerous, microbes start growing due to water in the fuel tanks and feed off the partially degraded fuel and any additives. Given enough time the microbes soon turn to slime and sludge. This microbial sludge destroys surface coatings, corrosion, clogging of fuel filtration, blockage of fuel supply, and malfunctioning fuel gauges. Organisms growing in fuel have been recorded since 1895 and were reported on aviation fuel since the early 1950s. One of the first known microbial attacks was in the wing tank of a Lockheed Electra aircraft in 1961. Since then there have been hundreds of reports touching on the correlation between microbes in fuel and corrosion, engine failure, and pipe fouling to name some. There are many tests and labs used to find the microbes in fuel and evaluate how to eliminate them which includes very harsh chemicals or disposing of the contaminated fuel entirely. Using Raman Spectroscopy with a concatenated laser system, we could preemptively test fuel for water content before the microbes have a chance to grow. This would cut down on time, money, chemicals, and wasted fuel.

2.2 Project Objectives

Through Raman spectroscopy we can find a chemical fingerprint of the sample by the wavelengths it emits without damaging the sample. It has already been used to find the chemical fingerprint of fuel through a single laser system this provided a way to correctly characterize and quantify the adulterant and chemical compositions in fuel. Usually, quality control on fuel is done by measuring the physical properties like density, vapor pressure, and boiling points rather than accurate chemical characterizations. It was tested and proven that Raman Spectroscopy with a laser (785 nm wavelength) was able to quantify the adulterant in each fuel sample with low errors. Water concentration can be measured with the Raman system by expanding the measurement spectral range into molecular stretch regions. To do this we must add a second laser system (680 nm wavelength) to see spectral regions where stretch regions are prevalent then concatenate the data from both wavelength spectrums. Not to mention, we can also do this in reverse. We can use this same system to detect fuel contamination in water sources. We could track pollution in the oceans, or even see if there are traces of leaked fuel in our drinking water.

2.3 Related

Currently, on the market, there are ways to test for water content in fuel. However, these are usually qualitative measurements using chemicals or dyes and heavy manipulation of the fuel sample, this means that once the fuel has been sampled it is no longer good to use. These kits often also take a certain amount of time before giving you the results in the form of a visual separation or color. This is not very accurate and would not be able to tell you exactly how much water is present or if there are any other chemicals in the sample. For example, the DS-1 test kit will check for the bacteria in your fuel which happens when there is water present for a certain amount of time. This DS-1 Kit takes 24 – 36 to see if there is a reaction on the sample. This does not provide quantitative data and it is not a quick process. There is also a W-5 test which provides a simple pass or fails result when adding a chemical to the test tube of fuel. If these chemical encounters water, they will turn pink. This one kit with five tests costs $46, not to mention the fuel is no longer valid to use once this chemical has been dropped in the sample.

As of right now, Innovative Photonic Solutions (IPS) has a patent out on a Raman probe that uses concatenated lasers. Under the patent, they are using dual-wavelength Raman probe byways of two laser sources. One of the excitation wavelengths is used for Raman scattering in the fingerprint region and the other has an excitation wavelength causing scattering in the stretch region. This is very similar to the RSCL however this is for a Raman probe whereas I am creating a Raman subsystem where a vial of the sample will be dropped into the system. As well as it is not just the optical Raman subsystem but also the integration of an algorithm in the computer to tell you what you're looking at spectra-wise, whether the amount of water detected in the fuel is dangerous or not.

2.4 Specifications

This section will look at the engineering and marketing specifications. This will provide a quick and easy way to see exactly what our goals are and a standard to hold the project to in senior design 2. We expect to meet all these specifications, we are not using these as stretch goals.

2.4.1 Marketing Specifications

There are many specifications and requirements needed to help keep the system safe, accurate, easily usable, and feasible. To compete with the current technologies, the device will be more accurate than the test kits provided on market right now for testing water content in fuel. The use of Raman spectroscopy with a dual concatenated laser system will provide reusability of only using one test system that in the long run will save money compared to standard kits. This will not only help the waste caused by other kits in the environment, but it will yield a yes/ no answer to if there is water present in the sample. Usually for these other kits like the ones mention previously, they are not environmentally friendly, nor do they have one kit that “does it all” in a quick fashion. The device will read if there are other chemicals present in the sample, while not using extra dyes or chemicals added to the sample for the reading. The algorithm and computer system in the later sections will provide data analysis that can’t be obtained from standard test kits. Most of all we want this device to be extremely safe as we are handling a substance that is easily ignitable with a high-powered laser. The device and system will conform to safety standards and all chemicals/ fuel will be handled safely. The prototype design will include an algorithm that will take the collected data from the spectrometer and determine if there exists any containment of water. The algorithm will be written in a way that it can easily be expanded upon further to add the ability to predict other contaminants and recommend a possible treatment to the sample to have it safe for use. The cases of safe for use can also be expanded upon to determine the safety for what sort of commercial device it can be used upon (i.e. an airplane versus a car).

2.4.2 Engineering Specifications

As for the engineering specifications, the device will have two lasers with wavelengths at 680 nm and 785 nm. The lasers will use fiber optics to connect to the system. The 785 nm laser will provide Raman excitation to the fingerprint region of the sample for the spectrometer. The 680 nm wavelength will also have Raman excitation, but it will be for the stretch region of this sample. The laser system will have its temperature-controlled box as well as a cooling system. As for the samples of the fuel, the samples will go into the system through glass vials. The device will be 80% accurate or greater. With a setup time of fewer than 10 minutes. The prototype design will contain a completely autonomous subroutine program that will execute immediately upon starting the device. This subroutine will be the top-level program that controls all other programs and interactions underneath it. This autonomous program will control all interactions with the device save for interactions that require human interaction such as collecting samples and storing data. Once these inputs have been taken from the user the subroutine program will immediately continue to run with the inputted information. This program will also take any values needed by the graphical user interface and deliver them, and vice versa from the graphical user interface to the main program itself. The autonomous subroutine will require debugging features to verify that certain steps are working throughout the design process, as well as the ability to add to the design at later stages in the professional product design (i.e. post prototype stages). The prototype design will include a custom graphical user interface to allow the user to interact with the design and allow more dynamic control over the design and testing experience.

The graphical user interface will present the user with the ability to collect a sample of data, collect calibration data, see the raw data right in the design space itself (i.e. on-screen), and export the data for future use. The graphical user interface will also allow the user to refine the way they want to collect the desired data by changing the wavelength size, the average amount of scans, and the sample time length. The graphical user interface will be operated on a touch screen and will be coded in a way to achieve this. The design will be created at first as a rough draft with a graphic design software capable of creating "layers" of the design, to export each piece of the design as its image. This will allow for easy access to the individual pieces both for editing and for coding, as well as offer the ability to add upon the design in the future. Finally, the graphical user interface will display the data in a clear, readable, and user-friendly manner, making it so that users will require a base amount of experience to operate the device and interpret the data being collected and presented.

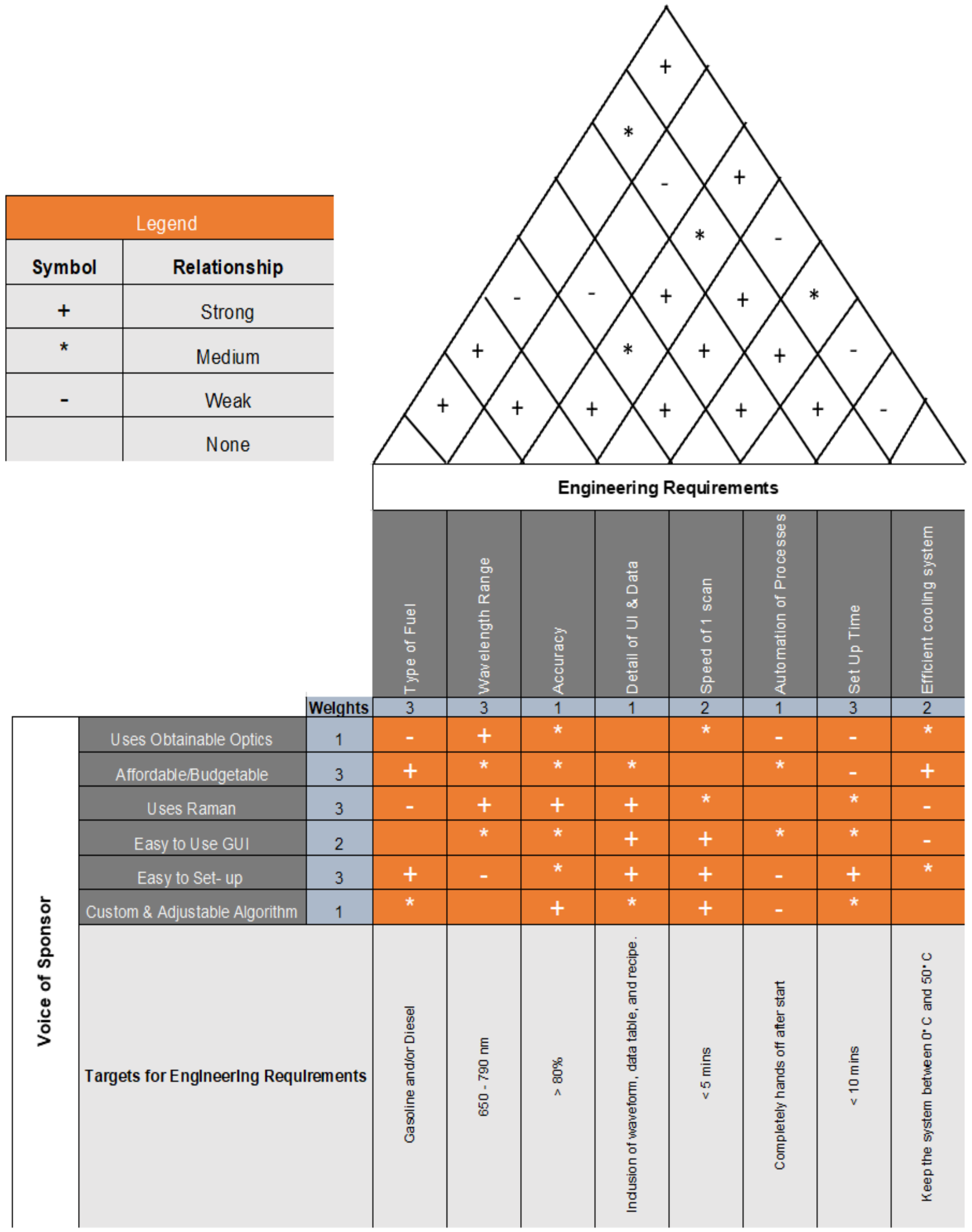
In the table below, our engineering specifications have been laid out in a simple chart format to make sure we stay on track and it is easy to spot what we are trying to do with the Raman spectroscopy with a concatenated laser system. The specifications highlighted in the orange sections below are the top three specifications we will try to meet with our demonstration in senior design 2.

Specifications	
Wavelengths	680 nm and 785 nm
Max Laser Power	350mW – 450mW
Integration time	1-10 seconds
Total bootup time	5 minutes
Total time for 1 sample	30 seconds
Accuracy (min)	80%
Samples	Kerosene, Gasoline, Diesel
Temperature	0°C - 50°C
Computer Interface	Raspberry Pi, Touch screen attached
Computer Connection	USB

Table 1. RSCL Specifications List

2.4.3 House of Quality

Representing the way our marketing specifications and engineering specifications meet, we can use a House of Quality diagram. It shows where each marketing specification relates to each engineering specification through a symbol that shows how strong the relationship between the two is. This is located on the legend, as seen in the figure below.



Legend	
Symbol	**Relationship**
+	Strong
*	Medium
-	Weak
	None

Voice of Sponsor	Weights	Type of Fuel	Wavelength Range	Accuracy	Detail of UI & Data	Speed of 1 scan	Automation of Processes	Set Up Time	Efficient cooling system
		3	3	1	1	2	1	3	2
Uses Obtainable Optics	1	-	+	*		*	-	-	*
Affordable/Budgetable	3	+	*	*	*		*	-	+
Uses Raman	3	-	+	+	+	*		*	-
Easy to Use GUI	2		*	*	+	+	*	*	-
Easy to Set- up	3	+	-	*	+	+	-	+	*
Custom & Adjustable Algorithm	1	*		+	*	+	-	*	
Targets for Engineering Requirements		Gasoline and/or Diesel	650 - 790 nm	> 80%	Indusion of waveform, data table, and recipe.	< 5 mins	Completely hands off after start	< 10 mins	Keep the system between 0˚ C and 50˚ C

Figure 1. House of Quality Diagram with marketing and engineering requirements.

2.5 Block Diagrams

In this section, we have provided block diagrams to better understand each section of this RSCL project. Including the optical, software, and overall hardware diagram. This provides a much better understanding of what each member of our team did.

2.5.1 Optical Block Diagram

The optical block diagram is seen below to demonstrate how the design for all the optics will be set up. It starts with light emitting from the laser and hitting each optical component on the way, we can see the beam path in orange.

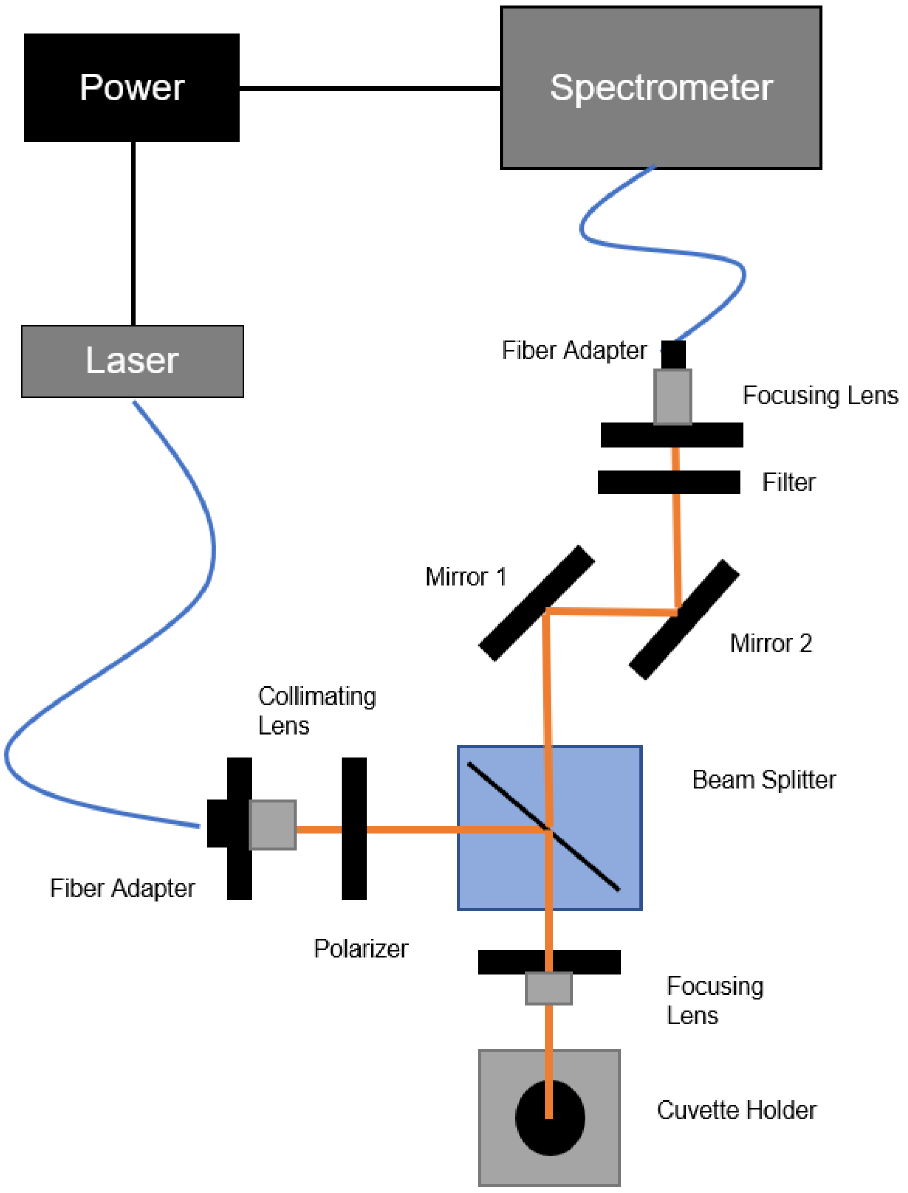


Figure 1.Optical Diagram, Yaelle Olivier PSE.

2.5.2 Hardware Diagram

Here is a Hardware Diagram representing our prototype, where each system is represented by a block with lines leading to other systems describing how they influence each other.

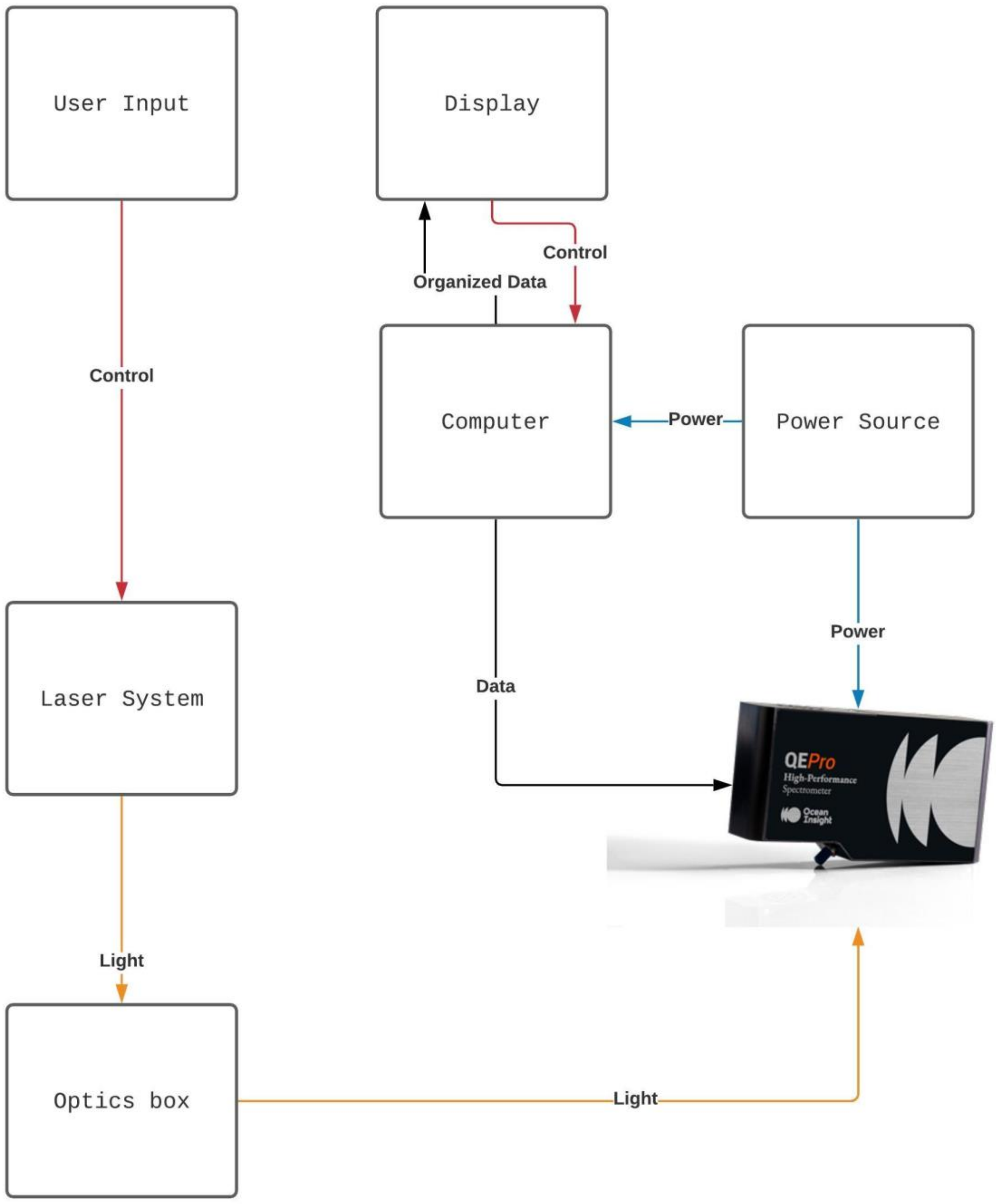


Figure 2. Hardware Diagram, Austin Collins CPE.

2.5.3 Software Diagram

Below is the flow chart that describes the software flow of the design. It is important to note that these steps will most likely not be exact and are subject to change. However, this will give a general idea of the design from start to finish, start at the moment the device is powered on and finish at when it powers off.

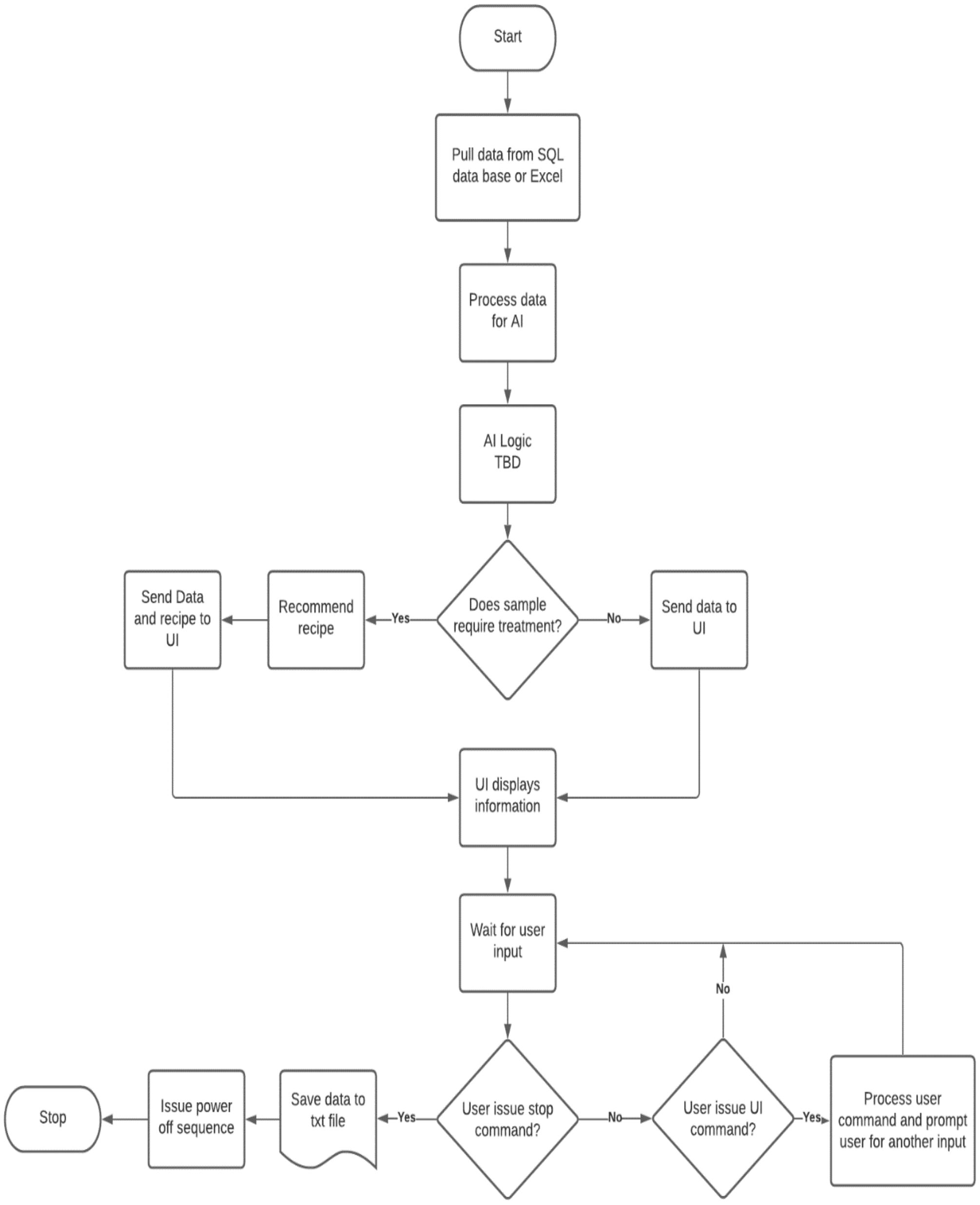


Figure 3. Software Diagram, Tyler Morris CPE

3. Research

In this research section, we will look at the background of each part of the system and why it works the way it does. We will discuss Raman spectroscopy and the role of a dual-wavelength system. While also looking into the way fuel works. This section will act as an introduction to the different concepts of the system before building it.

3.1 Raman Spectroscopy

Before looking at Raman spectroscopy it is important to understand what spectroscopy is. Spectroscopy in the simplest definition is the study of the interaction between light and matter as a function of wavelength. The light collected after the matter or sample is then used in a dispersive element to disperse the different wavelengths which are later collected by a detector as seen in figure 5.

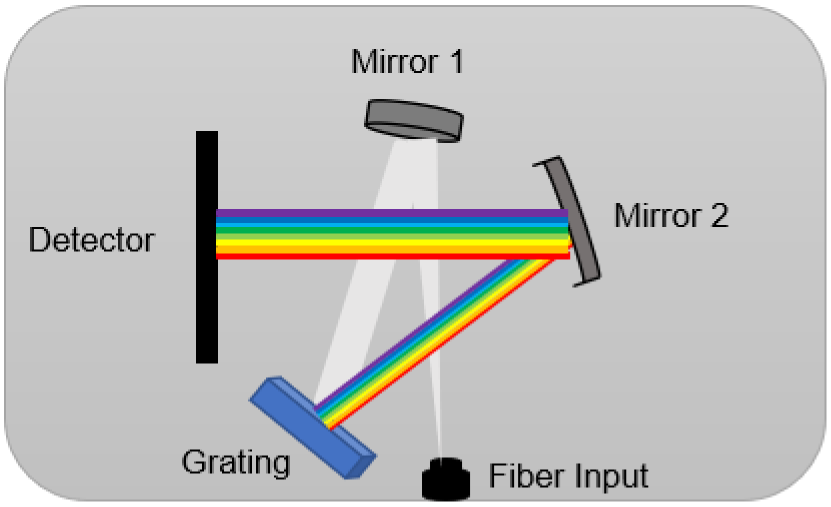


Figure 4. Inside look at a spectrometer

The earliest form of an example of a dispersive element is the glass prism where it can transform white light like the sun into a spectrum of colors like a rainbow, here we describe each color as a wavelength. White light is white not because of lack of color but due to mixing all the wavelengths, this is called additive mixing. As the white light goes through the prism each wavelength will experience a different frequency and velocity through the glass material. The light then exits the prism as a separated spectrum through the process of refraction and each wavelength takes a different angle causing the classic rainbow order. However, using a prism wasn't always the most useful as there was always a lot of wasted light. So, through the evolution of technology, it was possible to make a similar dispersive element that used reflection instead of refraction, so no light was missed when used for analysis.

This optical element is called a diffraction grating, it looks like a mirror but has thousands of narrow lines running through the glass surface undetectable by the human eye. Depending on what kind of light you are using or trying to measure will determine how many lines are in the grating, usually marked as a certain amount of lines per millimeter, the higher the lines per mm the larger the dispersion is. We can see this in the diffraction grating equation (1). Where d is the distance between lines, θ is the diffraction angle, m is the order number (m = 0,1, 2...), and λ is the wavelength.

$$d * sin(\theta) = m * \lambda \tag{1}$$

From this equation, we can see that the larger the line number, the smaller the angle will be. Depending on what you are trying to see through the detector at a certain wavelength will determine what kind of grating and line per millimeter you will use. Every spectrometer has a light source a grating, and a detector. The detector's job is to catch the light of the grating and digitize it into data considering the intensity of each wavelength.

Here we are using Raman spectroscopy, this form contains the same overall components of any spectrometer as previously explained. Usually, Raman spectroscopy is used with a single monochromatic light source, typically a laser that will shine onto a sample and will vibrate the atoms within the sample. This vibration will cause light to scatter, this scattered light is what we use to go into the grating and back into the detector. Most of the time the light is scattered in the same wavelength as the light source and it doesn’t provide useful information. This light is called Rayleigh scattering, we will add filters into the system to cancel out this scattering. We are looking for the Raman scatter this is the tiny amount of light that is scattered off the sample at a different wavelength than the source dependent on the chemical structure of the sample.

The next question would be how exactly does this scattering happen? To figure this out we need to look at our sample at an atomic level. Thinking of the compound as atoms or balls that are all joined by springs for example. We can get the idea that these molecules can vibrate and move a bit but not detach from the springs when the molecule is met with energy. This energy is described as a specific frequency that will initiate the vibrational mode. We know that frequency is the inverse of the wavelength, so to rephrase, at a certain wavelength the vibration will start to happen in different modes. It is important to note that this vibrational energy is not the same as a laser exciting a molecule. The photon that we see is not a real transition between energy levels. In Raman spectroscopy we see virtual energy states, so we see Rayleigh, stokes, and anti-stokes scattering. As seen in figure 6 below, Rayleigh scatter has no exchange of energy, so the incident light and scattered photons are the same wavelength or frequency. Stokes scattering is when the atom virtually absorbs the energy and the scattered photon has less energy when emitting a wavelength than the incident photon. The anti-Stokes scattering happens when the scattered photon has more energy than the incident photon. As said previously the Rayleigh is usually ignored and canceled out by filters and the anti- Stokes is usually very weak and ignored as well.

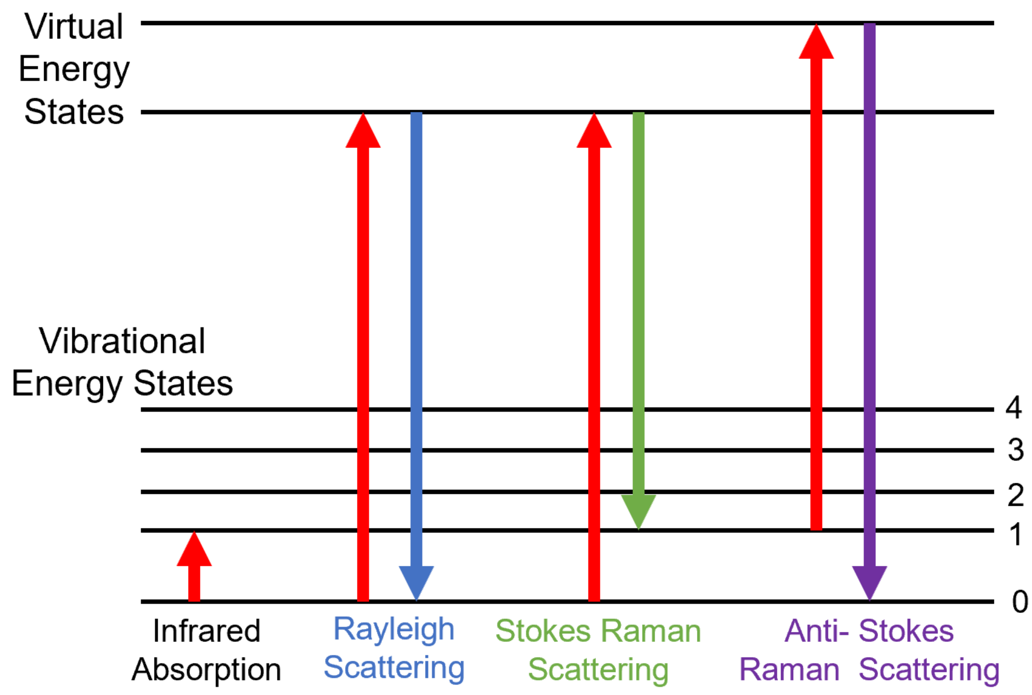


Figure 5. Representation of Rayleigh Scattering, Stokes, and Anti-Stokes virtual energy states.

For Raman shift as seen in equation (2) is usually reported in wavenumber and wavelength are inverses of each other. In the optics community, we use wavelength while in the chemistry community they use wavenumbers. A wavenumber is just 1/ λ usually in cm^{-1}.

$$\Delta\tilde{\nu} = \left(\frac{1}{\lambda_0} - \frac{1}{\lambda_1}\right) \tag{2}$$

Where the $\Delta\tilde{\nu}$ is the Raman shift, λ_0 is the excitation wavelength, and λ_1 is the Raman spectra wavelength.

3.1.1 Raman spectroscopy vs. other types of spectroscopy

Spectroscopy is generally used for the processes by which information about the molecular structure of an item is obtained through analysis of the absorption, scattering, or emission of electromagnetic radiation by the item.

Most organic compounds appear colorless because of the high-energy radiation which makes up the ultraviolet and visible section of the electromagnetic spectrum. This is due in part to the electrons in sigma bonds of the organic molecules requiring wavelengths of higher energy sources to disturb them to a higher energy level. However, electrons in Pi bonds may be promoted to a higher energy level by ultraviolet and visible light, with Ultraviolet and visible spectroscopy being able to observe useful information of molecules that contain Pi bonds. Highly colored naturally occurring organic compounds contain extensive systems of Pi bonds that are conjugated. Pi bonds are said to be conjugated when multiple Pi bonds are separated from each other by single bonds. Ultraviolet and visible spectroscopy provides incredibly detailed information on molecules that contain conjugated Pi bonds.

The atoms in organic compounds are described as being bonded to each other via Sigma bonds when the bonded atoms are bound together by the mutual attraction of the shared electron pair between the two. The two atoms' distance between each other is not static though, they vibrate to and from along with an average separation distance, the average bond length. The movements during this process are called stretching vibrations. The line joining two bonded atoms may sway back and forth within the plane it shares with another bonded pair of atoms or even bend to and from outside the plane, this movement is referred to as bending vibrations. Stretching and bending vibrations overall represent separate energy levels of a molecule, with the energy differences corresponding to energies of wavelengths in the infrared region of the electromagnetic spectrum. An infrared spectrophotometer makes infrared light traverse through an organic molecule and creates a spectrum containing the amount of light that is transferred on the vertical axis and the wavelength of the infrared radiation on the horizontal axis. The wavelength region 7 to 25 µm is known as the fingerprint region as the absorption pattern is unique to each organic structure. This makes infrared spectroscopy very useful for displaying the types of groups present in organic molecules.

The absorption of long-wavelength low-energy radiation, particularly in the radiofrequency is of the electromagnetic spectrum, is caused by the atomic nuclei in molecules. Most atomic nuclei having a small magnetic field influencing their behavior to become like tiny bar magnets. After being placed in a strong external magnetic field, nuclei will change to different energy states, with the simplest of cases having only two energy states. In a lower energy state, the nucleus’s magnetic field will align with the external magnetic field, while in the higher energy state it aligns against the field. The difference between these two energy states further depends on the strength of the external magnetic field, which ranges between 1.4 to 18 Tecla’s while also being irradiated with radio-frequency waves, in modern Nuclear magnetic resonance spectrometers. The energy difference between the two states of the magnetic energy level of the nucleus is measured as an absorption peak. Therefore, Nuclear magnetic resonance spectrometers show the most structural information of all the spectroscopic techniques, as the environment around the absorbing nucleus in the molecules affects the absorbed radiation.

Proton magnetic resonance spectroscopy measures the absorption peaks in organic compounds, which displays a lot of information about the molecular structure due to the hydrogen atoms. These atoms also absorb the energy of various wavelengths if the bonding environment is suitable. These nuclear magnetic resonance absorbances are displayed on the graphed spectrum as sharp peaks. The height of each peak correlates approximately to the strength of absorption, with no vertical scale on the spectrum, however. The horizontal scale on the graphed spectrum corresponds to the absorption of the protons in the compound tetramethyl silane $(CH_3)_4$. Which is an inert liquid that is added to the compound currently being analyzed, where its 12 hydrogen atoms are absorbing at a single position and creates a single peak, which is assigned the locational value of zero. This provides a frame of reference for the other peaks that appear on the spectrum, with the hydrogen atoms in the molecule that is under scrutiny appearing to the left of the reference peak, because it absorbs higher radiation energy than the tetramethyl silane's hydrogen atoms. The three key features of the proton magnetic resonance spectrum; chemical shift, relative peak size, and spin-spin splitting, provides information about the location and quantity of hydrogen atoms in a molecule. Combined with carbon-13 magnetic resonance, the structure of the molecule can be ascertained if the molecular formula is known.

Mass spectrometry displays information that is not dependent on the absorption of electromagnetic radiation but instead converts the molecules to ions and then separates them based on their masses. A chart then displays the masses of these ions with a measure of their relative quantity known as a mass spectrum. From these masses and clusters of the peaks in the spectrum, the exact mass of the molecule can be determined as well as details of its molecular structure. In a mass spectrometer, the compound being analyzed is inserted, vaporized, and passed into an ionization chamber, where it is battered by beams of high-energy electrons. These beams create a positively charged molecule, a molecular ion, that removes one electron from the molecule. These molecular ions then break into smaller fragments and are sped up by an electric field and stream into a mass analyzer containing a strong magnetic field. With the ions passing through the magnetic field, their path is changed into a curved path that depends on their charge and mass which pass by a detector recording their intensities and masses of the ions. The resulting mass spectrum displays the mass-to-charge ratio along the horizontal axis and the ion abundance along the vertical axis. Ions with a single positive charge of 1, the horizontal axis shows the corresponding mass of the fragment directly.

With these types of spectroscopy in mind, Raman spectroscopy is the only one that still satisfies the conditions for our device. Ranging from vaporizing the original compound or tainting it with other chemicals, defeating the purpose of the device, each of these analysis methods ends up in some shape or form destroying the original compound that was being analyzed.

3.2 Dual Wavelength Laser

As explained in the topic previous, usually we use a single excitation laser for a Raman Spectroscopy system but often these measurements will suffer from fluorescence. For Raman scattering the name of the game is to avoid fluorescence as much as possible, therefore near-infrared is preferred, as fewer molecules absorb near-infrared light especially compared to the visible region. That's the reason we are using a 785 nm wavelength laser because it provides a high Raman intensity and considerably low fluoresces. This wavelength provides us with our fingerprint region. For our excitation wavelength at 785 nm the wavenumber range will go from 0-2000 cm^{-1} . With fluoresce however it makes it difficult to see the "stretch region" in the spectra like 2000 to 4000 cm^{-1}. That is why we can integrate a second laser with a shorter wavelength and higher photon energy to collect the stretch region as seen in figure 7. For this, we want a second laser that is about 2000 cm^{-1} away from the first wavelength like a 680 nm laser. The lasers would be going through the same fiber and virtually hitting the same spot on the sample. Using the spectrometer to collect data from each wavelength we then concatenate the data together which will show an increased peak/ peaks in the stretch region. This has been used before and is patented by IPS (Innovative Photonic Solutions). However, it is patented for a Raman Probe. We can learn a lot from this moving forward and understanding exactly what each wavelength does.

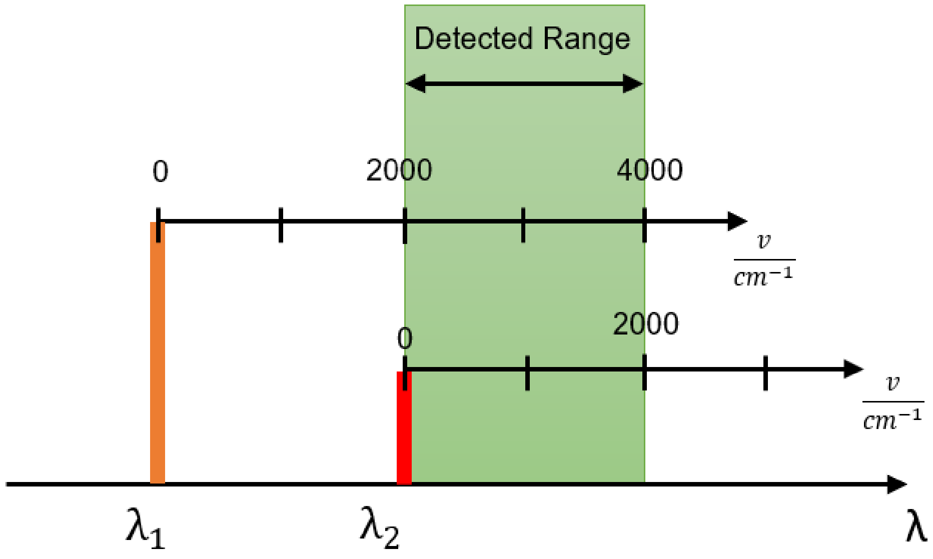


Figure 6. Detected range region where λ1 is 680 nm and λ2 is 785 nm

.

3.3 Fuel

There is a range of microorganisms that grow in fuel like bacteria, yeast, and mold at all stages of distribution and use, these microbes require water for growth. This is mostly found in storage tanks like seen in figure 8. At first, they are not able to be seen by the human eye until they have collected so much that they will turn into a slime that can get not only in the fuel tank but move into gauges and go around the engine. These microbes can directly cause metal corrosion and if found in a large quantity it could be dangerous for the person cleaning them out. These are present in many kinds of fuels, diesel fuel users have even coined the term "diesel bug" to describe the microbe growth epidemic in boats, motor craft, cars, trucks, trains, and power generators.

However, regular gasoline or diesel out of a pump is more likely than not already contaminated with other chemicals or water. We know this because it is a problem we are trying to solve. To start testing our prototype it is a better idea to start with a simpler version of fuel components like kerosene, dodecane, and hexane.

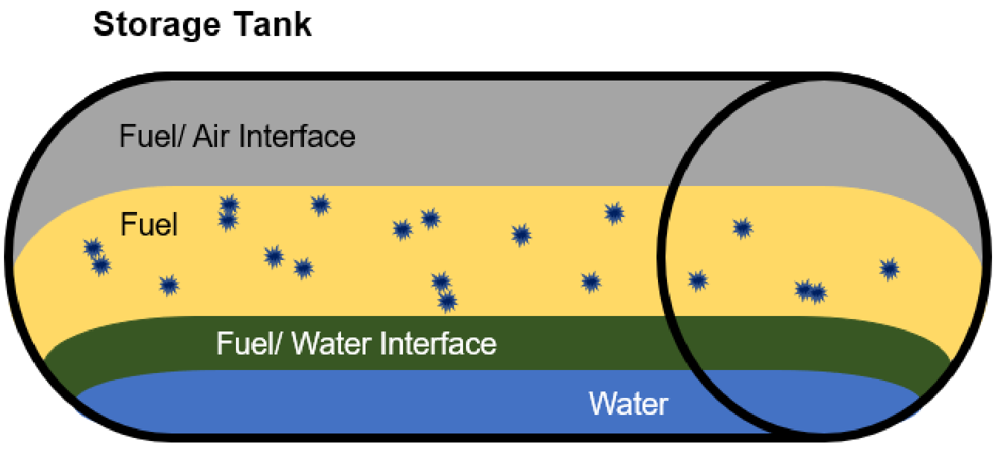


Figure 7. Storage tank with water contamination.

An important component in choosing our testing fuel will be its flashpoint. Which is the temperature at which the substance generates a flammable vapor. As well as its polarizability, which is the measure of how easily an electron cloud is distorted by an electric field. The molecule's polarizability affects the intensity of the Raman scattered radiation given by equation 3.

$$I_R \propto \nu^4 I_o N \left(\frac{\partial \alpha}{\partial Q}\right)^2 \tag{3}$$

Where I_o is the laser intensity, N is the number of scattering molecules, v is the frequency of the laser, α is the polarizability of the molecule, and Q is the vibrational amplitude. Therefore, the higher the polarizability of the molecule, the stronger the intensity of the resulting scattering, which will make it easier to analyze. Polarizability is generally affected by the size of the molecule, where the larger the molecule is the higher its Polarizability. Therefore, several three-dimensional models of the compounds and hydrocarbons are included to further demonstrate the size differences between the options and ergo their Polarizability.

Another important consideration that needs to be taken is the flashpoint of the fuels we desire to analyze. As the flashpoint is the lowest temperature at which a liquid will form a vapor in the air near its surface that will briefly ignite on exposure to an open flame. While a liquid is below its flashpoint, not enough vapor is produced to allow combustion but being above its flashpoint allows combustion to occur.

Especially when testing is taken into consideration where we have a potentially unstable prototype it is best to consider safety.

Kerosene, also written as kerosine, is a flammable hydrocarbon liquid that was historically produced by distilling crude oil but is more modernly produced through the process of fractional distillation of petroleum. Its usage as fuel is wide-ranging as it used as a component of aviation fuel, albeit in a highly refined form called RP-1 in which case it consists of over 60% of it, to the simpler uses of heating, cooling, and lighting, it is even used as a solvent for greases and insecticides. Kerosene comes in two grades: 1-k which contains less than .04% of Sulfur by weight, and 2-k which contains .3% Sulfur by weight. With that in mind, we would use 1-k as it has fewer impurities to taint the Raman readings. The flashpoint of kerosene is above 100 degrees Fahrenheit therefore making it much safer to test and store fuel than gasoline, whose flashpoint is –49 degrees Fahrenheit. Since Kerosene is a mixture of hydrocarbons, there is no specific chemical formula for it, however, there is a general representation for it: CxHy, where x ranges from 6 to 16 and y changes if its alkane or cycloalkane. Often the molecule for dodecane is used as a model of a "typical" molecule of kerosene as seen in figure 9.

Dodecane is a component of gasoline and is primary used as a solvent and is also used in jet fuel research. Dodecane is an alkane hydrocarbon, meaning that it consists of hydrogen and carbon atoms oriented in a tree structure where the carbon-carbon bonds are single. The molecular formula for Dodecane is $C_{12}H_{26}$ meaning it is a decently sized hydrocarbon (as demonstrated by Figure 4), meaning its Polarizability is quite decent, marking it as a good potential test fuel in the creation and calibration of our device. With its Flashpoint being a high 165 degrees Fahrenheit, it makes it an extremely safe fuel in testing as well as storage.

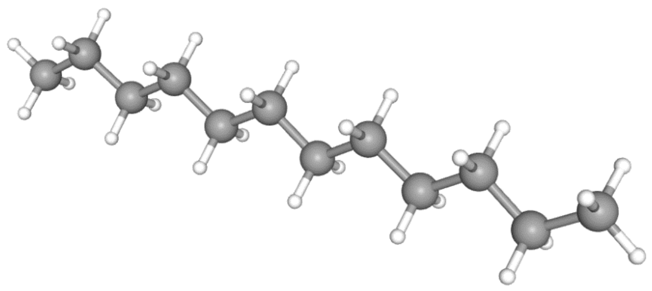


Figure 8. kerosene molecule reprinted with permission from PubChem

Hexane is an alkane, which is a type of hydrocarbon that contains only single bonds and as such is made up of nothing but hydrogen and carbon. They are also known as saturated hydrocarbons as they contain the greatest number of hydrogen atoms per carbon atom. It is typically found in raw natural gas but is removed before shipping to be used as an energy currency. Its flashpoint is –9.4 degrees Fahrenheit making it an extremely volatile liquid and needs to be handled with caution. With the molecular formula of C_6H_{14} (Figure 10) its polarizability is on the lowest options in terms of our testing fuels. However, even with its volatility and smaller molecular structure, it is easily sourceable at Ocean Insight making it an easy access test fuel.

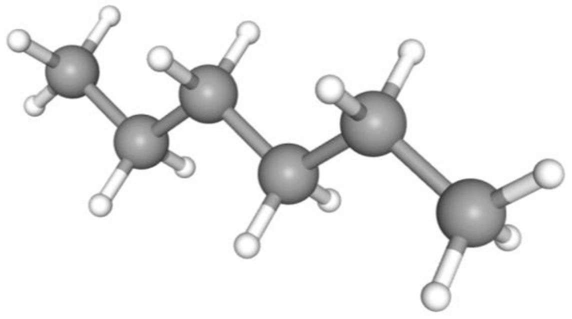


Figure 9. Hexane Molecule reprinted with permission from PubChem

As diesel fuel is a combination of many different hydrocarbon molecules there is no one specific molecular formula for it, however, it ranges from between 12-20 carbon in the molecules. The flashpoint of diesel fuel is around 130 degrees Fahrenheit making it safe for storage and testing fuel. However, as it is difficult to find a source of diesel fuel that does not already have a small concentration of water contaminating it, it is not the most ideal testing fuel. But on the other hand, its great amount of carbons in its hydrocarbons makes its Polarizability the greatest of the potential testing fuels that we have seen so far. Provided that we could find an acceptable source of pure and uncontaminated diesel, then diesel fuel has by far the most potential as a testing fuel that we have seen so far.

3.4 Other Consumer-level Tests

The DS-1 Fuel test kit is one of the main fuel test kits that test for some of the major contaminants of fuel, which are bacteria and mold. It is a double-sided stick that when dipped into the fuel sample will display the bacteria results on one side and the mold results on the other. Where the Bacteria side will display results within 24 to 36 hours after exposure, and the Mold side will display the results within 36 to 48 hours after exposure. The results simply being, if there are no spots on either of the sides being a negative result for the presence of any kind of contaminant and any sort of reaction resulting in a positive presence of the contaminant.

While cheap these tests come with many downsides, primarily the wait time for test results. When testing for contaminants the sooner that the result is known, the sooner one can find the solution to the problem. The other problem that is presented is that the concentration of bacteria or mold in the fuel is not indicated with the test results, therefore creating more legwork for the consumer as they must find out the concentration of the contaminant to properly treat the fuel. Finally, the tests do not tell the user what the contaminants specifically are mere that they are present.

Another fuel test kit is the FCT-100 which tests the fuel’s water content and clarity. Designed a simple pass or fail the test, if the 5 indicator marks are visible then the fuel passes the clarity test, and if they are not then it fails. Then the powder is added to the sample to test for water content. If the water content is 200-500 parts per million, then the powder will contain pink spots, but if the parts per million are higher than 600, then the powder turns hot pink. If the powder turns hot pink, then the test is failed, however leads to the contamination of the test sample, which is something that should be avoided if possible.

A more basic test for checking the water content in fuel is the basic separation test. But this test isn’t a good indicator as to the quantity of water contaminating the fuel source but does indicate that there is water in the fuel. It is also another test that requires a long period of waiting for the separation of the fuel and water to occur.

The other alternative testing method would be to mail the samples to an external lab for testing. While it is a hand-off and requires less effort, it also presents its unique downsides. Primarily the fact that the sample must be shipped to the lab. This not only increases the amount of waiting time for the results but also presents the possibility of the sample being damaged while being delivered to the testing lab. The delay could be increased even more so by the possibility of the lab being backlogged.

Therefore, our proposed device ends up solving all the issues and problems that these other consumer-level tests of fuel contain. Where there is not a large delay between the application of the test and the results of said test, while also not contaminating the test sample.

3.5 Hardware/ Software Interactions

One of the main questions when designing a product such as this is the possible interactions between the selection of hardware and the selection of software. What sort of benefits can be gained from choosing one over another? What sort of disadvantages will there be in selecting a certain operating system versus another? This section will cover all the decisions made for these possible interactions, why these decisions were made, and how they affect the system and design process.

From the hardware aspect, the main decision in this space was what sort of microcomputer would be best to achieve the specifications and make the design process as smooth as possible. Given the nature of the product, the decision of using a higher power but more efficient single board computer versus a low cost, low power microcontroller was a given.

The tools needed to use to interact with the spectrometer could not possibly fit and run-on microcontrollers, let alone the finished program. Many microcontrollers also are limited in the programming languages they use, meaning that to use a programming language optimized for efficiency, the libraries of the microcontroller would have to be re-written in the desired language. So that points us in the direction of single-board microcomputers.

3.5.1 Choosing a Single Board Microcomputer

With the knowledge that the design needs something at least as powerful and flexible as a single board microcomputer, the next task would naturally be to determine which to use. There are many options to choose from and narrowing the selection down to a handful can be very difficult.

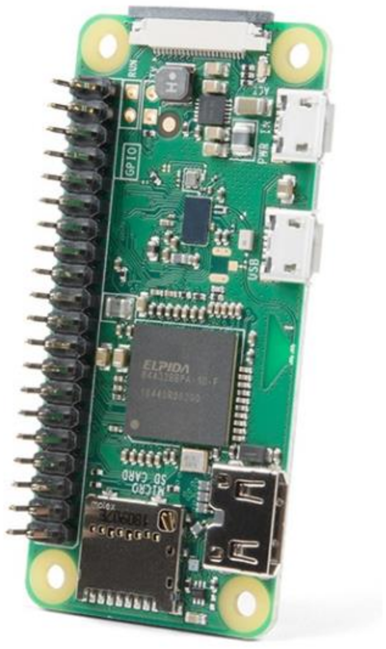


Figure 10 (Used with the permission of Helen Lynn). The Raspberry Pi Zero is an extension of the Pi Zero family that launched in early 2017. It has all the functionality of the original Pi Zero with added connectivity and more powerful computing power. It offers Linux capabilities as well as the NOOBS

One very enticing option would be the Raspberry Pi Zero W displayed in the figure above, with an attractive price of just $14, is smaller than a credit card at 65 mm long by 30 mm wide and coming from one of the leading brands in the embedded Internet of Things (IoT) companies, it seemed like a possible choice. What the Raspberry pi lacks though is computing power, at just 512 MB of Random-Access Memory, and a 1 GHz processor, it became clear that choosing this option would hinder the design process. Constantly having to work around these specifications while testing and debugging the design would be a nightmare. However, the option is one to keep in mind in the post software design phase of the design process (one that would be reached beyond the scope of this prototype design stage).

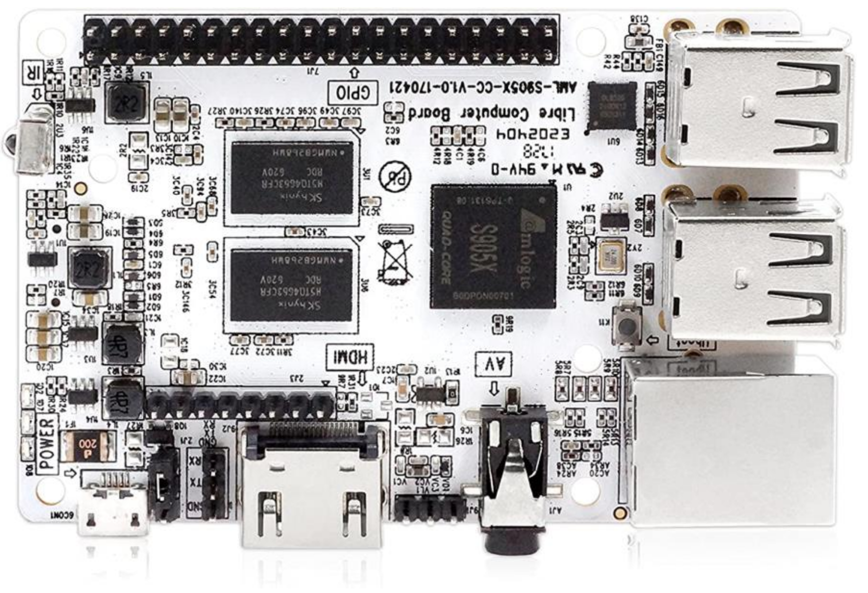


Figure 11(Used with the permission of Da Xue). The Le Potato is the main hardware platform of libre computers with Android and Linux support. It comes from a long-supported System on Chip family with a proven track record of mass deployments by large design companies like Google and Amazon.

Another possible option is the Libre AML-S905C-CC, otherwise known as the Le Potato displayed in the figure above. This board being Libre's flagship single-board microcomputer and designed to be a direct competitor with the Raspberry Pi 3 Model B, has a lot to offer. Being double the price of the Raspberry Pi, what it lacks in pricing, it makes up for in specifications. It has nearly twice the computing power of the Raspberry Pi Zero W, which would make up for the design roadblock that would be testing the software components of the product. The board is however much larger than the Raspberry Pi Zero W coming in at 4.8 x 3.03 x 1.06 inches. However, being this design stage is the prototype design stage, the size of the board can be neglected for the most part, as after the software design is optimized in future stages, the size of the board can become one of the smaller boards available on the market. The main drawback of the Le Potato is the fact that it is optimized for Android programming. While the software could be translated to work with an Android operating system, it wouldn't be the easiest to design given the nature of the desired specifications.

With the comparison of these two boards, the main deciding factor to determine which single-board microcomputer to use comes down to what makes the design process the smoothest. Which board gives us enough processing power and RAM to complete our prototype testing and debugging. After all, that is what this stage of the design process is about. Optimizing and condensing can be taken care of at a later stage once the initial problems of the project are identified and solved. At a later stage and in the interest of minimizing cost inboard, materials to house the board, future-proofing, and the other aspects of achieving the highest cost to performance ratio, the recommendation would be to go for the Raspberry Pi Zero W.

3.5.2 Latte Panda

Due to hardware factions discussed in the Hardware Parts of this paper and the reasons described while comparing the previous two boards, the option decided for the prototype design software phase is the Latte Panda.

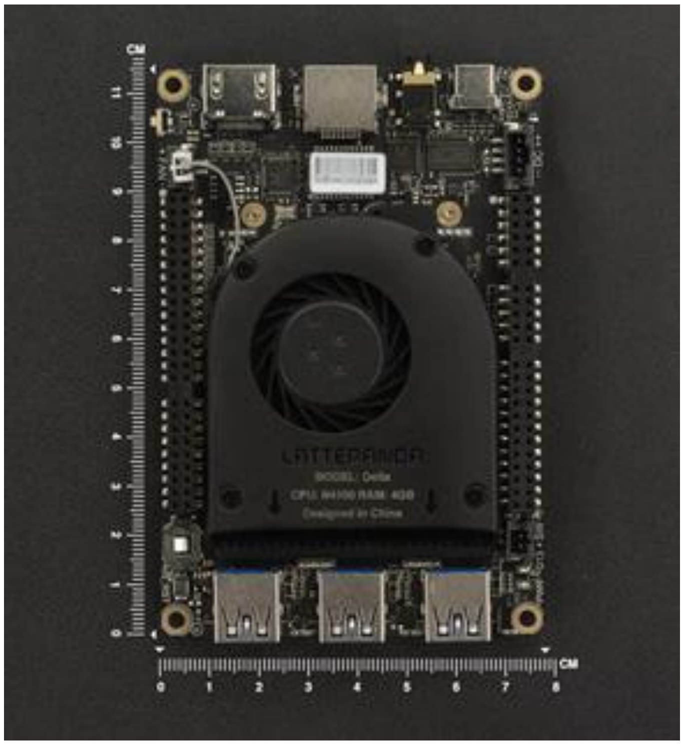


Figure 12(Used with the permission of Cain Zhang). The Delta 432 is Latte Panda's main choice from the Internet of Things devices or Artificial Intelligence cores. It supports both Windows 10 and Linux operating Systems with a solid list of specifications and design choices. It aims to be the smallest and thinnest computer while still holding enough computing power for larger projects like those in Artificial Intelligence and Machine Learning.

The Latte Panda Delta 432, displayed in the figure above, while having a high price of $188, does seem to be the best option for finding something that can work within the initial design process, but also be able to scale down to a smaller, cheaper, and lower-powered option. The board sports a 2.4 GHz quad-core Intel Celeron N4100 processor with integrated Intel UHD graphics, 4GB and DDR4 Random-Access Memory, built-in wi-fi connectivity, three USB 3.0 ports, and an HDMI output port. The processing power will be more than enough for this prototype; however, the board could be used to implement attractive features such as Machine Learning in future design stages. The wi-fi connectivity, while not used in this design, could be used to communicate with a SQL database or some other form of massive data storage/cloud data storage. This could be useful in implementing and managing the potential Machine Learning algorithm discussed earlier, which would be able to learn and identify new forms of contamination in fuel samples, and using the recipe algorithm, recommend treatments to allow the fuel to be safe for usage again. The USB 3.0 ports are helpful as the lasers used in the design is USB-powered, so the need for an additional USB source to power these components isn't needed. Finally, the HDMI port is extremely helpful, as the recipe algorithm will be feeding information to a graphical user interface, which will be displayed on a screen for the user's benefit. The Latte Panda also supports both Linux and Windows operating systems fully, allowing the choice of two of the three most popular development operating systems (please refer to the software selection section).

To conclude this section, the Latte Panda is the best choice for this prototype design phase, having the processing power needed to develop a workable algorithm and graphical user interface, the connection ports needed to cut down on excess hardware and power supply options, and the ability to future proof and add more features to the design later in future design process steps.

	Zero W	**AML-S905X-CC**	**Delta 432**
Brand	Raspberry Pi	Libre	Latte Panda
Price	$14	$50	
Size	65 mm x 30 mm	4.8 in x 3.03 in	96 x 60 x 14.5
RAM	512 MB	2 GB	4GB
Processor Clock	1 GHz	1.512 GHz	2.4 GHz
Operating Systems	Windows macOS Ubuntu	Linux	Windows 10 Linux
Extra Features	Mini HDMI Port Micro USB Powered	4 x USB 2.0 Type A HDMI 2.0 Micro USB Powered 100 Mb Ethernet	External Memory Support Wi-Fi Connectivity 3 x USB 3.0 Type A 1 x USB 3.0 Type C HDMI 2.0

Table 2.Comparing different Single Board Microcomputers

3.5.3 Analog to Digital Conversion

Analog to Digital converters converts continuous-time and continuous-amplitude analog signals into discrete-time and discrete-amplitude digital signals, making the data easier to process and store, while also making the data more accurate and reliable by minimizing the number of errors. Computers, as well as any digital processor, need the data to be in digital form for the preparation, translation, and storage of the data. After the analog data is converted into digital data, numerous calculations and manipulations can occur.

One of the main properties of the Analog to Digital Converter that must be observed is the resolution of the converter. The resolution shows the number of values that can be displayed over the range of analog values that are sent. Therefore, the resolution affects the value of the quantization error and affects the maximum signal-to-noise ratio for an Analog to Digital converter without using oversampling. The input values are typically stored in a binary form inside of the analog to digital converter, leading the number of discrete values to a power of two. For example, an Analog to Digital converter with a resolution of 12 can display an analog signal's input to one in 4,096 separate levels. The resolution also directly affects the accuracy of the resulting digital signal output, as the bit length increases, the more the level changes more closely resemble the original analog signal.

To improve the performance of Analog to Digital converters, dither is sometimes introduced. Dither is a minuscule amount of random noise that is introduced to the input before being converted. This serves to randomize the change in voltage required to change the output code level based on the signal. As opposed to the signal ending at low levels, the range of signals that the analog to digital converter can change is extended, but at the cost of increasing noise. This mainly helps to increase the resolution of the sample but does not always improve the accuracy.

3.5.4 Types of Analog to Digital Conversion

Successive Approximation Register analog to digital converters balance speed and resolution while also having the capability to process a thorough range of signals. As the technology has existed for quite some time, the current Successive Approximation Register designs are on the relatively cheaper side while also having a great amount of reliability. The average Successive Approximation Register analog to digital converter makes use of a sample-and-hold circuit that receives an already manipulated analog voltage from the front-end. An included Digital to Analog converter makes an analog benchmark voltage equal to the resulting digitized output of the sample. Both then input into a comparator that conveys the resulting comparison to the Successive Approximation Register. This is repeated as many times successively as the bit resolution of the analog to digital converter until a close match to the input signal is discovered. However, the Successive Approximation Register Analog to digital converts do not contain anti-aliasing filtering, so if a low sample rate is picked, false signals are introduced by the Successive Approximation Register.

Delta-sigma Analog to Digital converters improves upon the amplitude axis resolution while reducing the high-frequency quantization noise that is in Successive Approximation Register converters. The design of the Delta-sigma converters allows for more varied uses that require as much resolution as possible. These converters over-sample the analog signal much more than the chosen sample rate, they then create a high-resolution stream of data from the over-sampled data at the previously selected rate. This makes a much higher resolution data stream while creating the opportunity for multistage anti-aliasing filtering, nearly eliminating false signals in the digitized data stream. This however slows the process of digitization, making it slower than Successive Approximation Register converters.

Dual Slope Analog to Digital converters while accurate, is also on the slower side of the converters. The idea is that the conversion from analog to digital is using an integrator. With the voltage being input and being allowed to ramp up for a time. Then a voltage of the opposite polarity is put in and is ramped down back down to zero. After reaching zero, the system determines the value of the input voltage by comparing the run-up time with the run-down time, and what the reference was. While reliable, the amount of time to complete the process is more substantial than the previous two. There is also a trade-off between the speed and resolution, as a high value in both cannot occur.

Flash Analog to Digital converters, also known as direct-conversion converters, operate quickly and with almost no latency. Therefore, making them the most choice form of analog to the digital converter when the highest sample rates are desired. The conversion between analog and digital signals occurs by comparing it with a known benchmark value. As more benchmark values are used in the conversion, the higher the accuracy of the result. If a 12-bit resolution is used, then the inbound analog signal is compared with 4,096 known values. As the resolution is increased, the larger the Flash Analog to Digital converter becomes, and the sample rate must consequently be decreased.

Pipelined Analog to Digital converters are primarily used when sample rates are high enough that the Successive Approximation Register and delta-sigma Analog to Digital converters cannot be used, but the speed of Flash Analog to Digital converters is unnecessary, pipelined Analog to Digital converters are used. In Pipelined Analog to Digital converters the analog signal is not grabbed by all the comparators simultaneously, and instead, the energy required to convert the data from analog to digital is spread. This results in higher resolutions without requiring a large amount of energy. However, the sample rates cannot be as high as the normal Flash approach and then making the latency typically 3 cycles.

3.6 Software Selection

This section will be discussing the decisions made in choosing an operating system, and a programming language to design the recipe algorithm. These choices were made due to a variety of factors such as the single-board microcomputer chosen, the ease of design, the strengths and weaknesses of each operating system and language considered, and the interest of future-proofing and making a transition to a low scale design for mass production more feasible.

3.6.1 Operating System Selection

First, the selection of an operating system needs to be made. With the selection of the Latte Panda (see section 3.5.2 Hardware/Software Interactions), the three operating systems become two. The three most popular operating systems for development are Linux, macOS, and Windows. Latte Panda doesn't support macOS innately, so in the interest of the design process is as smooth and as quick as possible, only Linux and Windows will be considered. The two are essentially equal in development aspects. Neither offers any pros or cons to the actual recipe algorithm design. Windows on one hand offer more programs that could aid in the design of the prototype; however, it also sports a decently high price tag of $139. Linux on the other hand has many free distributions and has multiple flavors to choose from such as MX Linux, Linux Mint, Manjiro, and Ubuntu.

In the interest of keeping the software design step smooth, Windows will be selected as the operating system of choice. It offers compiler support for all the considered programming languages as well as an easy-to-use subroutine system that will be used to automate the system, requiring interaction with the user only when the user is required to input a command. However, in future design steps, with the consideration that a lower cost and lower power board will be chosen, a Linux distro may be considered to reduce cost. The algorithm itself will be flexible between the two and the only redesign in this event will be the subroutine process, which will be trivial with the Windows version present as a guide.

	Windows 10	**MacOS**	**Linux (Ubuntu)**
Price	$139	Free	Free
Supported	Yes	No	Yes
Difficulty	Easy	Moderate	Moderate

Table 3. Comparing Operating Systems

3.6.2 Programming Language Selection

Next, to determine the main programming language. With there being a vast number of options, the choice was made to narrow it down to three. The most supported languages in the industry and on the internet are C++, Java, and Python. The choice to decide between the three is a bit more difficult, however. C++ offers scripting extensions with C# and can manage memory which Java and Python do not easily allow. Java also has scripting extensions with JavaScript and has a large amount of documentation due to being the most popular programming language in the industry currently. It also is very popular in graphical user interface design, which is one of the major aspects of software design. Python offers inherent scripting capabilities and is a rising language in the industry. It also is a white-space programming language also adds to the ease in programming with it.

This alongside the fact that it is an extremely flexible language is very attractive. This is due to the fact there are many open-source publicly available libraries that can be used and incorporated into any design. To add to all of this, Python is a rapidly growing language for Artificial Intelligence and Machine Learning applications.

	Python	**C++**	**Java**
Pros	Flexible Growing Light Weight Expandable Scripting extensions Popular with ML	Popular in the industry Moderate support Memory management Scripting Extensions (C#)	Industry-standard A large amount of support Easy to make UIs Scripting extensions (JavaScript)
Cons	Newer Less Documentation and Support Harder to make UIs	Less flexible Must manage memory No UI capabilities	Must edit the design with each update Growing older/outdated Not very flexible

Table 4. Comparing Programming Languages

Considering all the above, Python can be determined to be the best language to design the programming aspects of the software side of the computer stack. Many libraries that act as drivers for the possible spectrometers listed have been written and are available to the public. Graphical user interface design is also possible with Python; however, it is not as easy as it is with Java. This one con is vastly outweighed by the pros, being flexibility, algorithm design speed, and potential future applications of scripting use and Artificial Intelligence/Machine Learning Applications.

3.6.3 Development Environment Selection

Finally, with the designation of Windows as the operating system and Python as the programming language, the last step is to choose a development environment to make the process of designing the software components go as smoothly as possible. The two most popular selections for this are Anaconda and Chocolatey.

Anaconda is a free open-source distribution of the Python programming language. It is aimed towards scientific computing and comes with many tools such as JupyterLab, Spyder, and PyCharm. All these tools can help in the creation of the software components needed and add their pros to the whole that is the distribution. They also allow the generation of Python scripts for a clean and easy programming experience.

Chocolatey is a developer-centric package manager. The goal of Chocolatey is to add, update, uninstall, and maintain packages in the background so the user doesn't have to be so hands-on with updating and maintaining the components necessary for a design. It is designed to be used with a text editor, with the most popular choices being Atom and Notepad++.

For this project and the potential future applications and additions, Anaconda will be chosen. The assortment of tools each tailored to handle certain tasks with ease, that other languages like Java or C++ would struggle with, is too attractive to pass up. It has both Windows and Linux distributions and allows for a seamless transition from the design phase to the scripting and testing phases. This also considers the potential implementation of Artificial Intelligence and Machine Learning later in the design process, but this is outside the scope of the prototype design process.

3.7 Choosing the Display

For selecting the User interactivity aspect of our Design, it comes down to whether we should use a touchscreen display or a regular display with a mouse and keyboard. Both come with their own set of benefits while greatly impacting the portability of the device. If we were to choose the regular display with a touchpad and keyboard it would allow for more customizability with our programming and allow for more functionality to be added to the device overall. Allowing for a larger display for ease of use and visual clarity, while not sacrificing the functionality of the GUI. However, if we choose this option, we sacrifice the portability aspect that is desired.

With the touchscreen alternative, it allows for a much easier design and creation process for the casing of our device as there is only the touchscreen to take into consideration. Allowing for a more compact design, allowing for better portability, and helps accomplish the goal of field use for the device. The use of a touchscreen also increases the speed of use of the device, as users can promptly clicking anything with the use of a finger or stylus pen instead of having to scroll around with a mouse and keyboard.

A great example of a touchscreen that would suit our needs is the Raspberry Pi Touch Display. With a size of 7," it is an ideal size to keep the finished device to a compact and portable size. And with a resolution of 800 x 480, it allows a clear display and should aptly display the GUI clearly and concisely as well. It also comes with its drivers to support an on-screen keyboard, which allows for more functionality and flexibility with our program and GUI.

When it comes down to it, the touchscreen is the clear winner. As it helps satisfy our goal of portability and in-field use, while also maintaining ease of use and functionality of the device. While also not sacrificing the quality of the display, touch screens have made great advances in quality.

3.7.1 EDP vs HDMI

Embedded Display Ports is a display panel interface standard meant for portable and embedded devices. Based on the VESA DisplayPort Standard, and is a high-performance external audio/visual interface, with the capability of displaying resolutions of 4K and beyond. The embedded DisplayPort cables also allow for higher transmission speeds compared to the more traditional Low-voltage differential signaling cables. While also using fewer conductors in cable design which reduces the amount of space and possible cost if the design allows for it. Many laptops, all-in-ones, and high-resolution tablets have resorted to using EDP due to its convenience and ease of use. It was mainly developed since the majority of DC-coupled high-voltage wing interfaces require the use of external interface chips or process changes that compromise the performance of the system, this includes low-voltage differential signaling cables, digital visual interface cables, and High-definition multimedia interface cables. Embedded Display Ports also features the ability to better manage system power for longer battery life, lowers the electromagnetic interference and radio frequency interference allowing for a sleeker and less bulky design for the desired interface. The most convenient aspect of Embedded Display ports is the fact that all the power, data, and control is input through one single assembly, therefore reducing the clutter of cables inside the device while also requiring fewer connections than low-voltage differential signaling cables.

High-Definition Multimedia Interface is the leading digital video, audio, and data interface connecting high definition displays throughout the great range of electronics on the market. High-Definition Multimedia Interface cables have a great variety of sizes and types. Type A, which is the standard size, typically used in TVs, DVD players, video game consoles, and home theater receivers. Type C, which is the miniature size, is primarily used on Digital single-lens reflex cameras with one end having a standard size to connect to a computer, TV, or video projector. Type D, which is micro-sized, is used on portable devices like smartphones and tablets with a Type-A connector on the opposing side. With a single cable, it can physically transfer video and audio signals digitally from source to video display. However, this means that a separate cable must be used to power the display that we use in the device, leading to more clutter and cables to manage.

While both options are great, we have opted to choose an Embedded DisplayPort style display. As the all-in-one convenience for both power and video in one cord while simultaneously saving on power usage, makes it much better for the portability aspect of the device.

3.7.2 Different Types of Touchscreens

There are five types of touch screen monitor technologies that are the most popular now. They are Resistive Touch screen, Surface Capacitive Touch screen, Projected Capacitive Touch screen, Surface Acoustic Wave Touch screen, and Infrared Touch screen. Each having its list of advantages and disadvantages when compared to the other.

5-Wire Resistive touch screens are the most common ones in use. It is comprised of a glass panel and a film screen, each being covered with a thin metallic layer, and are separated by a narrow gap. Upon user touch, the metallic layers connect which results in electrical flow. This is detected by the voltage change. This allows it to be used with almost any object, causes it to have low power usage, makes it resistant to dust, oil, and other liquids. But this causes it to have lower image clarity and makes it vulnerable to damage from sharp objects and scratching.

Surface Capacitive Touch screens contain a transparent electrode layer on top of a glass panel that is covered by a protective cover. When a finger touches the screen, it reacts to the ability of the human body to absorb electricity and some of the electric charges are transferred from the screen. This decrease is detected by sensors at corners of the screen with the controller using that to determine what area was touched. This makes it so that the screens are durable, highly resistant to dust, oils, and other liquids, and scratch-resistant. However, this makes it require bare fingers and is highly sensitive to electromagnetic interference and radio frequency interference.

Projected Capacitive touch screens are nearly identical to Surface Capacitive Touch screens but can be activated with surgical or thin cotton gloves. As well as having the capacity to allow the simultaneous use of two or more fingers. It is comprised of a sheet of glass that has transparent electrode films and an integrated circuit chip embedded in the glass, creating a three-dimensional electrostatic field. When a finger touches the screen, the ratios of the electrical current change, with the computer being able to then detect the touchpoints. This allows for a more resistant screen, resistance to dust and liquids, and multi-touch. But is still sensitive to electromagnetic interference and radio frequency interference as well as still being limited to finger usage.

Surface Acoustic Wave touch screens use piezoelectric transducers and receivers that are placed along the sides of the glass plate in the monitor. This makes a grid of ultrasonic waves along the surface. A portion of the wave is absorbed upon touch, which the receiving transducer uses to locate the touchpoint. This allows for better scratch resistance than projected or surface touch screens as well as great clarity of images. But means that it cannot be used with hard items, like a fingernail, water can cause false triggering, and solids, such as dust, create non-touch areas until removal.

Infrared touch screens use infrared emitters and receivers, creating a grid of light beams across the screen that cannot be seen. When an object crosses the light beam, the sensors locate the touchpoint. This makes it have the highest image clarity of all and is impervious to surface scratches and enables the use of multiple touchpoints. However, this means that particulate buildup on the screen/frame could impede the light beams which will cause malfunctions, is sensitive to direct high light interference and has a higher cost.

	Latte Panda	**Raspberry Pi**
Resolution	1024 x 600	800 x 480
Transmittance	85%	N/A
Surface Hardness	6 H	N/A
Response Time	≤ 16 ms	785
Tapping Durability	200,000,000 times	N/A
Support	Latte Panda Alpha and Delta products	Raspberry Pi products
Connector	10 cm FPC Extension Cable	GPIO port and DSI port
Size	7.1 x 4.3 x 0.2 inches	7 inches
Included drivers	N/A	Virtual keyboard
Price	$69	$73

Table 5. Comparing the available touch screens.

3.7.3 EDP Touch Display

The EDP Touch Display, boasts a high resolution of 1024 x 600 and is compatible with the alpha and delta line of Latte Panda boards, but not their other ones due to the different CPU architectures and the positioning of the circuit elements. An additional feature of this Touch Display is the fact that it is designed for Human-Machine Interface scenarios and is therefore designed to prevent dust from entering the screen and reducing the reflection of light. Its compatibility with Latte Panda alpha and delta boards allowing for the customization of screen power, brightness, and contrast allowing for even more customization of the screen to better fit the GUI. Although the display has special double-sided tape on the edges of the screen, we are going to custom 3-D-print backing to help reinforce its attachment to our device as one of the goals is for it to be portable. Therefore, making sure that the display is adequately secured to the case while also being able to maintain a stable connection to the single board microcomputer is a must. A picture is included to demonstrate the clarity of the display, while not sacrificing the quality with its compact and sleek size (Figure 14).

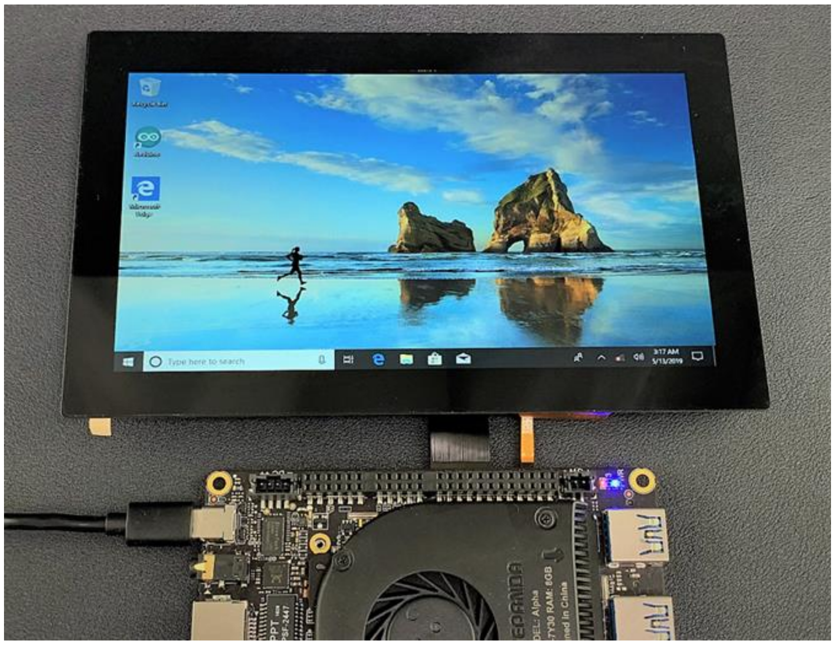


Figure 13. EDP Touch Display. Reprinted with permission from LattePanda

3.7.4 EDP Touch Display secure backing

Although the EDP Touch Display by Latte Panda includes double-sided adhesive tape around the screen edges to help secure it to our desired case, we are not going to have that be the only thing securing it. Therefore, we measured the dimensions of the Touch Display and designed, and 3-D printed a secure backing for it to make sure that it stays in place and won't displace or fall. Additionally, a custom-made rubber gasket is placed between the edges of the screen and the case to seal the edges of the device to prevent any kind of unwanted liquids, debris, or other particulates from getting into the device while not interfering with the operation of the touch screen. The model is shown in Figure 15.

Figure 14. Placeholder image for gasket and secure backing. Reprinted with permission from Barnwell.

3.7.5 Titan Case

As the device we are designing will ideally be used in the field, protection of the single board microcomputer is paramount. The microcomputer cannot be left exposed to particulates that can appear in the field, whether it be dust or dirt, the fact that the case is made of high-strength plastic not only helps in preventing any static damage from occurring but also protects the microcomputer from any kind of damage from possible collisions inside of the case. Protecting the microcomputer helps minimize the failure of the device as it is central to its operation. As we are selecting the Latte Panda Delta 432, an easy solution presents itself in the form of the Titan Case (figure 16). By also making use of the Titan Case we can minimize the amount of vapor that could reach the microcomputer through either leak from the testing vials, or through the careless bottling of the fuels into the vials. Even with the addition of the case, the microcomputer remains nearly the same height, this allows us to maintain variety in terms of placement for the microcomputer's location inside of the prototype, while not negatively impacting the protection the case can provide to the microcomputer. The case does present a problem with the fact that it is made of plastic where it increases the chance of overheating as plastic does not dissipate heat, however, provided that our air flow quality is high enough the problem is mitigated fairly easily.

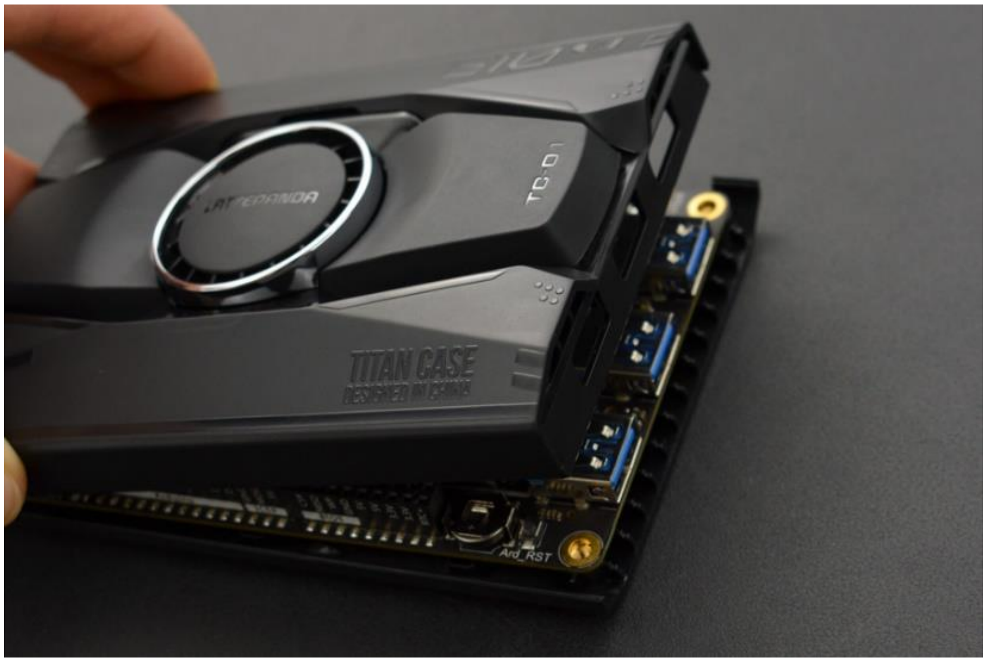


Figure 15. Titan Case, Reprinted with permission from LattePanda.

4. Design Constraints and Standards

In this section, we will go over the related standards and constraints that need to be considered. Standards are understandably one of the most important parts of a design this is where the project meets safety and documentation for the government or other engineering organizations will come into play. These standards are implementations for safety and to avoid liability problems. These are like specifications as they are a guideline for the developers to use when creating a new product. We also need to think about constraints here when considering the new project and how it may affect the work later. Like time, safety, environment, and manufacturability.

4.1 Related Standards

Looking at related standards, most are pulled from the Institute of Electrical and Electronics Engineers (IEEE) Standards Association and the American National Standards Institute (ANSI). These are accredited associations that professionals and students can use when dealing with electronics, information technology, laser, and more.

4.1.1 Optical Standards

One of the most important parts of the RSCL is the laser system. Raman spectroscopy only catches a fraction of the light produced to turn into data. This means that it is crucial to use higher-powered lasers to get better data. However, the higher power of the laser the more dangerous it is, and it comes with its own set of standards. As seen in the figure below this is the kind of sticker you see when working with a laser. This is an image of the exact laser we will be for this project. On this sticker we are given a lot of information, usually, you will learn this information from a laser safety class. When classifying lasers, the range is class I to class IV with class I being the least hazardous and class IV being the most hazardous. For example, of what each laser class entails, class I is usually a laser printer or CD player. Class II and IIa are like a barcode scanner at a supermarket, class IIIa products are like laser pointers, and class IIIb and IV are industrial lasers or research lasers. In the case of this project we are looking at a class IIIb laser, according to the FDA, these lasers emit between 5 mW and 500 mW output power, these are legally not allowed to be used as a laser pointer or demonstration laser products. These are dangerous lasers and can cause either temporary or permanent eye injuries. According to the FDA standard d 21 CFR 1040.10 and 1040.11) “requires a warning label on Class IIIa and IIIb products. Class IIIb products must also have a key switch and connector for remote interlock. The products are also required to have identified and certifying labels and instructions for safe use.”

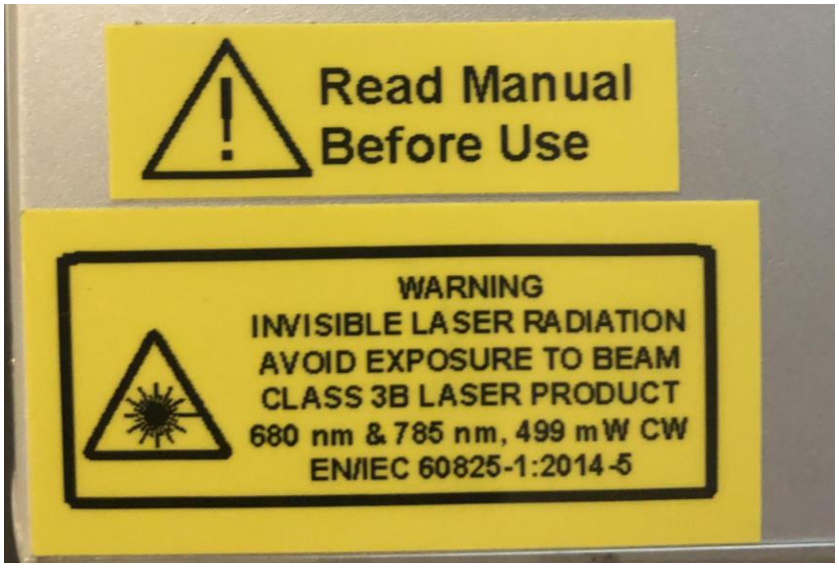


Figure 16.Laser warning sticker on Fat Boy Laser from IPS.

For more laser safety standards, we can look at ANSI Z136.1 American National Standard for the Safe Use of Lasers: 2000. According to ANSI, this safe use of lasers is for class IIIb and class IV lasers. They provide a practical means for establishing a safety program to protect the user from beam hazards and non-beam hazards. Where beam hazards are potential injury to the eye and potential injury to the skin. Non-Beam hazards include electrocution, fire, exposure to air contaminants, hazardous gases, laser dyes, and solvents. To avoid these hazards, they have implied that these lasers must be in protective housing, warning signs, and label requirements. Users must follow standard operating procedures (SOPs) and wear personal protective equipment (PPEs).

Not only does the laser have standards but using a fiber optic does too. According to ANSI Z136.2 "(safe use of optical fiber communication systems utilizing laser diode and led sources), This standard addresses the hazards of, and guides the safe use, maintenance, service, and installation (manufacture) of optical communication systems (OCS) utilizing laser diodes or light-emitting diodes (LED) operating at wavelengths between 0.6 µm and 1 mm, and not intended for visual communications."

4.1.2 Hardware/ Electrical Standards

The UL 796 Printed-Wiring Motherboard Standard creates the safety requirements for printed-wiring motherboards. The tests included in this international standard include the flammability of the board, its bond strength, the construction of its conductors, the evaluation of the dielectric materials, and the conductors used in the creation of the motherboard. It also details the silver migration test.

According to UL 796, the flammability classification will follow the classes outlined in UL 94. The board cannot receive a greater flammability classification than the base material, or the uncoated samples if coated samples are tested. However, before the testing, the specimens are subjected to thermal shock conditions. This means that there is no wrinkling, cracking, blistering, or loosening of any conductor or delamination of the materials or bonding layer because of thermal shock. All the test samples are adjusted to a temperature of 250 degrees Fahrenheit for one and a half hours before being subjected to the thermal shock tests.

The temperature limit and a greater time can be changed if specified by the manufacturer. First, the bond strength test is conducted by subjecting the test samples are soldered or another equivalent operation at the maximum temperature/dwell-time limits set by the fabricator. If a soldering process involves repeated soldering operation with a range of intervening cooling periods, the minimum cooling period is used. If required, a removable solder resist is applied at the time of testing to make sure that the solder does not bond to the sample, the solder resist is removed before further testing is conducted. The samples for the flammability tests must be 5 inches long and .5 inches wide with the minimum thickness to be used in production. These samples then go through the same production process as the represented board, minus any conductive material. If a coating, is going to be used in production, another set of specimens containing the applied coatings must be supplied.

Thermal Cycling procedure A is the process in which the sample is thermal shocked at the manufacturer's set maximum temperature and time, and are then subject to a thermal conditioning process for three cycles following the schedule of 1) 48 hours at 18 degrees Fahrenheit above the maximum operating temperature set by the manufacturer, 2) 64 hours at a range of 91.4 degrees Fahrenheit to 98.6 degrees Fahrenheit with humidity ranging from 85 to 95 percent, 3) 8 hours at 32 degrees Fahrenheit, and 4) 64 hours at a range of 91.4 degrees Fahrenheit to 98.6 degrees Fahrenheit at 85 to 95 percent humidity.

Thermal Cycling procedure B is the process in which the sample receives a dielectric voltage withstand test at 1000 volts for one minute with a ramp rate of 50 volts per second, then are thermal shocked at the desired maximum temperature and time. Then the samples are subject to a thermal conditioning process for three cycles following the schedule of 1) 48 hours at 18 degrees Fahrenheit above the maximum operating temperature set by the manufacturer, 2) 64 hours at a range of 91.4 degrees Fahrenheit to 98.6 degrees Fahrenheit with humidity ranging from 85 to 95 percent, 3) 8 hours at 32 degrees Fahrenheit, and 4) 64 hours at a range of 91.4 degrees Fahrenheit to 98.6 degrees Fahrenheit at 85 to 95 percent humidity. After the three cycles, the sample is then subjected to another Dielectric voltage withstand test at 1000 volts for one minute.

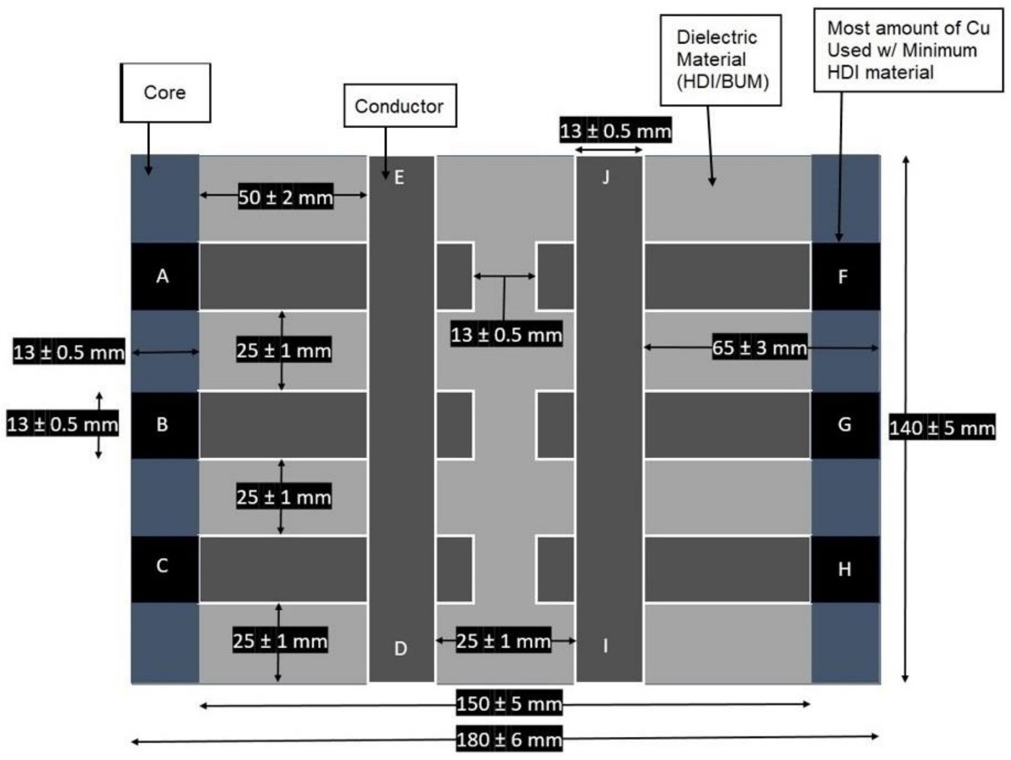


Figure 17. Displaying the required construction for side A

For the Dielectric materials that are used as insulation between traces that are not separated by any laminate material, complying with standard UL 746e, no evidence of cracking or delaminating when subjected to the Vertical flame test, Bond strength test, and Dielectric Crossover test. For the vertical flame test, twenty samples are acquired and are subjected to the test described in the previous section.

For the bond strength test, six samples that contain foil-type or clad conductors are subjected to the bond strength test. Three samples with foil-type or clad conductors are then subject to thermal cycling procedure A, which after said process the boards then go through the bond strength evaluation. Another three samples containing paste-type crossover conductors also go through thermal cycling procedure A and then go through the plating adhesion test. For the dielectric crossover test, three samples are constructed following the included designs (figures 18 and 19), these contain traces insulated by a dielectric material that goes through Thermal Cycling procedure B first.

Before testing begins, the sample is placed on an insulation material with a thickness of at least 5 mils and a dielectric strength of at least 1.5 kilovolts. And a 1000-volt Dielectric Withstand test is conducted on the insulation material before the test of each set of samples. Then following the testing outlined in UL 746a, Test Pointed 1 through 6 are tested, if any warping occurs, then test points 7 through 12 are tested. The test points are outlined in table 6.

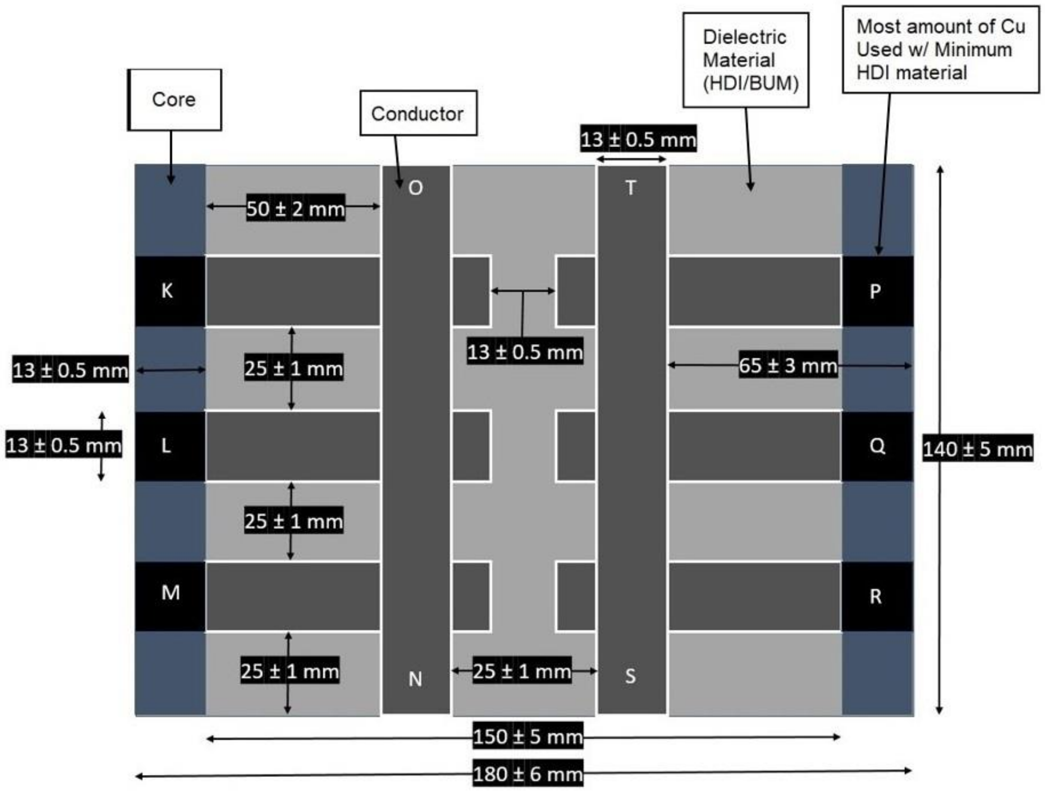


Figure 18. Displaying the required construction for side B

For the silver migration test, five test samples are required and if a permanent coating is used to slow silver migration another set of test samples must containing each different coating is made. The test samples are wired to allow adjacent conductors to represent the minimum spacing to be charged at a DC potential equal to the anticipated voltage rating for the board. If the product is going to be for AC use only, then an AC voltage equal to the anticipated voltage is applied. Before starting the silver migration test, a dielectric voltage of 1000 V plus twice the voltage rating asked by the fabricators is applied to the samples for sixty seconds, the voltage is increased by 200 volts per second until the final test voltage is obtained. Then a 1/8-amp non-time delay fuse is applied to the circuit to detect whether a short is caused in the circuit due to the migration of silver. The samples are then stored in a humidity chamber maintaining a temperature between 70 degrees Fahrenheit and 77 degrees Fahrenheit with a relative humidity of 95 to 100 percent and charged at the desired voltage rating for 56 days. At the end of the process, the samples are removed from the humidity chamber and maintain a temperature range of 70 degrees Fahrenheit to 77 degrees Fahrenheit, with a 50 percent relative humidity for 2 days. After this process each board is inspected for any signs of silver migration, then a second dielectric voltage withstand test is performed. If not, silver migration is detected and there is no dielectric breakdown, then the results comply, and a minimum spacing and maximum voltage rating are assigned.

Side	Test point	From	To
A	1	A	D
A	2	B	D
A	3	C	E
A	4	F	I
A	5	G	I
A	6	H	J
B	7	K	N
B	8	L	N
B	9	M	O
B	10	P	S
B	11	Q	S
B	12	R	T

Table 6. showing the test points needed for the Dielectric Crossover test

As stated in UL 796, about conductors, "Printed-wiring conductors shall be of etched, die-stamped, pre-cut, flush-press, or additive-type copper, copper alloy, aluminum, silver, or other conductive material having similar corrosion resistant properties." With the caveat that if silver is used it must be investigated for silver migration, which is the ionic movement of silver between two adjacent traces resulting in an electrical short. If any sort of solder or conductive coating is used, it must be "smooth, ductile, cover the conductor surface, and not interfere with electrical connection in the end-product assembly." The surface will be smooth and even while also not having any "wrinkles, holes, voids, blisters, corrosion, or other imperfections," that will cause malfunctions or impair the function of the board.

The UL94 standard specifically pertains to flammability ratings of the product and testing through three methods: surface burns, vertical burn, and horizontal burn.

For Surface burns, the device can be rated 5VA or 5VB. On 5VA surface burns, the burning stops within 60 seconds after five applications of five seconds each of a flame to a test bar. The specimens must not have a burn-through, making this the highest UL94 rating. The 5VB surface burn, burning must stop within 60 seconds after five applications of five seconds each of a flame to a test bar, the specimen may have a burn-through. For vertical burns, the devices can be rated at V-0, V-1, and V-2. Where V-0 devices stop burning within 10 seconds after two applications of ten seconds each of a flame to a test bar, with no flaming drips allowed. V-1 devices stop burning within 60 seconds after two applications of ten seconds each of a flame to a test bar, with no flaming drips allowed. V-2 devices stop burning within 60 seconds after two applications of ten seconds each of a flame to a test bar, with flaming drips being allowed. For the H-B horizontal burn rating, a slow horizontal burning is applied on a 3 mm thick specimen with a burning rate of fewer than 3 inches per minute or ceases burning before the 5-inch mark. This rating is also the lowest UL94 rating and is also considered to be "self-extinguishing." As the single board microcomputer will be purchased, we already know that it follows the requisite standards.

The plating adhesion test consists of applying a strip of pressure-sensitive cellophane tape that is ½ inch wide. The tape is then pressed onto the surface of the conductor pattern being tested by using a steel roller around 3 and ½ inches in diameter and 1 and ¾ inches wide, which weighs about 4.5 pounds. The roller is used to remove all the air bubbles created during the tape's application so that the tape's length on the pattern is a minimum of 2 inches. The tape is then removed by gripping one end and removing it at a 90-degree angle at a rate of 12 inches per minute. The tape used for the test must have adhesion of around 30 to 40 ounces per inch, which is determined by the Standard Test Method for Pressure-Sensitive Adhesive Coated Tapes Used for Electronic Applications, ATM D1000.

The tape is applied and removed from three separate locations on the sample with new tape used for each application. If there are no small particles of metal that adhere to the removed tape, it passes the test. However, if some small particles of the metal show on the tape, then it demonstrates that the sample does not have an acceptable bond strength and may also show that there is an overhang. The small metal particles being removed would be the protective plating or conductor pattern peeling off.

For a motherboard to pass the bond strength evaluation, after receiving the thermal shock test, the average strength of the bond between the material and the printed wiring must not be less than 2 pounds per inch of width of each of the individual strips of the conductor when tested using methods one, two, and four. As well as not be less than 1 pound per inch of width of each of the individual strips of a conductor when tested using methods three and four.

Testing method one consists of taking a uniform width of the conductor and peeling it from the base material for a distance of ¼ inch at the rate of around 12 inches per minute. The angle between the base material and the printed conductor must no become lower than 85 degrees. The force needed to separate the conductor from its base material must be measured, with three determinations to be made while using three test samples that contain: a conductor with the minimum width on the sample, a 1/16 inch wide conductor, the conductors of the other widths that have been specified by the fabricator, and an edge conductor. The average strength of a bond is determined for each of the widths of the conductors on each sample. A plated or separately formed contact must be tested, however, if it is constructed to be at least three times wider than the minimum printed line conductor on the printed-wiring board then the contact does not need to be tested.

Testing method two consists of taking two of the three samples from testing method one and then placing them in a full-draft circulating-air oven for 240 hours that complies with the Standard Specification for Forced-Convection Laboratory Ovens for Evaluation of Electrical Insulation, while being maintained at the temperature described by the formula of t_2= 1.076 (t_1 + 288) - 273, where t_2 is the oven temperature in Celsius and t_1 is the temperature rating of the printed-wiring board in Celsius.

Testing method three only occurs if the fabricator specifically requests it, where after testing method one takes place two of the three samples are placed in the full-draft circulating-air oven that complies with the Standard Specifications for Forced-Convection Laboratory Ovens for Evaluation of Electrical Insulation for 1344 hours. The oven is to be maintained at a temperature following the formula t_2= 1.02 (t_1 + 288) - 273, where t_2 is the oven temperature in Celsius and t_1 is the temperature rating of the printed-wiring board in Celsius. For testing method four, after the samples that go through testing method two or testing method three, the test samples are given enough time to cool to room temperature and are then subject to testing method one again.

After going through any of these testing methods, there should not be any evidence of wrinkling, cracking, blistering, or losing of any of the conductors or any delamination of the base material or the bonding layer after the thermal shock test or the oven conditioning test. If bond strength cannot be measured due to conductor embrittlement of unaged samples, the conductor needs to be evaluated manually by pulling up an end of the conductor and then pulling up an end of a conductor that has been through the oven conditioning test in the same manner, this compares the final and initial bond strengths of the conductors concisely.

During the construction of the motherboard, there are multiple compliances that the board has to be constructed to, where the samples should be able to represent the rest of the boards being produced. One compliance being the manufacturing process, as each sample must be produced using each of the steps of the most severe process in regards to time duration and temperature of any of the given steps. During the process of creating the conductor, it must result in smooth edges without any excessive undercutting with dimensions that are not less than represented by the test board. Any undercutting will not be greater than the conductor's thickness for each side. Chromic or sulfuric etchants will represent all of the other etchants, with any other acidic or alkaline etchants being representative of all etchants except for chromic or sulfuric. In the base, a flush-press metal conductor must be placed with at least 80 percent of the conductor's thickness. When there are no temperature differences, a change of imprinting method, for example, a silk screening to a photographic method, shall not always mean that a board needs to be tested. However, a retest is required if changes one through three occur. These tests will be the thermal shock and the bond strength test unless another test is specified. Change one is any change in any process if the temperature on the surface of the board exceeds 212 degrees Fahrenheit, or the maximum operating temperature of the board, whichever of the two is greater. Change two is a change in an etchant to a chromic or sulfuric etchant. The retest will follow the Thermal Shock and the Bond strength test, if the fabricator changes from any acidic to the alkaline etchant, or the other way around, to anything except chromic or sulfuric etchant, then retesting is not needed. Change three is the adding of plated-through holes to an existing process. The tests required will follow the plating adhesion test, with the addition of any other metallic plating, besides silver, that does not make contact with the base material not requiring testing. If changes one through three are not done in the original plant or at a foreign factory of another fabricator, then five tests must be conducted on the incoming products. Test one details that when the board is described for use at a temperature higher than 9 degrees Fahrenheit below the maximum temperature index of the base material, solder shock tests are conducted on the boards using the values described in the table below, or on the solder shock limits of the board's fabricator, whichever values are greater. The boards are then examined for delamination and blistering.

Base-material grade	**Time (seconds)**	**Temperature (°F)**
XXXP, XXXPC	10	475
FR-1, FR-2, FR-3	5	500
FR-4, FR-5, G10, G11, CEM-1, CEM-3, GPY	20	525

Table 7. Base materials with the time and temperature

Test two details that when the board is described for use at a temperature equal to or lower than 9 degrees Fahrenheit below the maximum temperature rating of the base material. The solder shock test described in test one is conducted and then followed up by a bond strength test of the minimum conductor width after aging. Test three details that when both the fabricator of the board and the outside processor use acidic or alkaline etchants that are not chromic or sulfuric, then no testing is required. The foreign processor will not use a chromic or sulfuric etchant if the board fabricator can use chromic or sulfuric. Test four details that plating that touches the surface of the base material requires testing as described in test one. Test five details that when tests one through three are needed, one sample out of a lot of 1,000 or less is chosen. When a failing test result is obtained on the sample, then double the original number of samples is tested. If another unacceptable result is obtained on any one or more of the new retest sample, then the entire lot is deemed unusable. The data is kept for at least 1 year to reference in the future.

4.1.3 Software Coding Standards

Something that needs to be considered when designing anything in the STEM world is standards. Standards ensure that designs will be safe, secure, and reliable to maintain a level of quality across the fields. This is required so when others go to use or work upon or with a design, they will be able to achieve their goals with little to no trouble. It is important to note the distinction that when this section talks about software standards, we are not referring to functional safety standards like IEC 61508 or programming safety standards like CERT, but more focusing on the guidelines that every programmable design needs. These guidelines are considered by the above standards as well as many others, however, we do not want to go into something as explicitly defined in this stage of the design.

Software Safety

The first guideline to look at will be safety. Designing a code to be safe is extremely important because in many cases the program or device is linked to other mechanisms or devices that control many important things. If a piece of code fails to be safe and is directly associated with a companies main database, the entire database could be wiped out. It is important to make sure that one does not take shortcuts that could endanger the integrity of the code. An example would be not backing up the data when manipulating it in the recipe algorithm. There is always a chance that the data gathered will not be able to be recovered if it were to be deleted, this adds more emphasis to adding a query database framework into the design, but that isn't the main focus of this topic. Following common safe coding guidelines to make code clean and usable will ensure that the program has no way to causing catastrophic damage. Making sure that the program halts all interaction with the system should something go wrong will make sure no false signals are sent. Maintaining defaults when variables are to be collected will ensure that users don't accidentally receive false information, or at the very least will be aware that something is wrong with the current test. There are so many techniques that to cover them all they would warrant their document or documents to list them. However, this section is more so about the acknowledgment of these techniques and informs the reader of the intent to use them in the prototype design.

Software Security

The next guideline is going to be security. Cybersecurity is a major threat in the information technology field, causing billions of dollars of damage each year. This topic will be split into two major points, software security, and the lesser considered hardware security. The hardware security portion of this document is listed in this section since many of the attacks relate to each other in a way that warrants doing so.

Software cyber threats have been a problem in the industry for many decades now. Cybersecurity experts and researchers are constantly discovering and combating new attacks that seem to appear daily. These threats are shut down by issuing patches to users. However, manufacturers can only do so much. In the end, it is up to the user to make sure they are keeping their device and designs secure from potential threats. A threat that immediately comes to mind when discussing the ideas offered in this prototype design is SQL injection hacks. These hacks are executed by taking advantage of a system using SQL queries to search databases to insert their data into the systems. If an attacker is smart enough, they can manipulate the searches in such a way that from the intended view of being able to search a database of chemical readings and recipe recommendations, they can insert their credentials and give themselves administrative privileges to the whole system. From here the damage they can inflict is limited only by the present worth and their imagination. These types of threats will need to be taken into account when designing the program as well as implementing possible ideas in the future. Another step that will be taken to make sure the device is safe is to keep operating systems up to date. If the device intends to be connected to the internet in any way in the future, it will need to maintain a level of security that makes sure nothing can be downloaded onto the device without the user's knowledge. This is however less of a threat due to the device being relatively locked down, and unable to interact with the rest of the device like a normal computer.

Hardware security is a growing issue in the last decade. New attacks that target the user's hardware have been developed, and the worst thing is they are not patchable like software threats. These threats are present when the device is designed and if overlooked and mass-produced, will be present in every unit that has been shipped. While some steps can be taken to ensure this design is maintaining this security as much as possible, many of these problems will be at the board manufacturer level, which is not a part of this design process, and not a field the sponsor company has any affiliation with. Still, the designers of this device have a duty to ensure the design is as safe as it can be. The first step that can be taken is to make sure the hardware is up to date. This can be costly and it generally out of the question to units already shipped in a finished design. However, these steps need to be taken to maximize security. Doing so will make sure the current parts are as secure as the manufacturer can make them. Another step that can be taken is to make sure each device and design are properly secured and handled, as attackers can appear from inside the design company and compromise the device. These steps and more can and should be taken to ensure the security as well as the safety of the device

Software Reliability

Ensuring a design is reliable is to make sure that the device functions the way it should every time. There will be cases where the device fails, especially during the prototype design stage, however steps should be taken to ensure that the software is as reliable as it can be. The major culprit here fails states, the designer should make it a goal to think of as many fail states as possible that the device could enter, and plan around them. Some examples that could be possible in this device are to hang when the device fails to receive a reading from the spectrometer, crashes when the device cannot save to a directory or fails to use a function because an invalid integer is given as an intensity level. All of these fail states and more can be predicted and can be worked around. The device should at the very least never crash, allowing the user to analyze what possibly could be wrong. Default states should be implemented to ensure that the code never has any case where it needs a variable and lacks it, and exception handling like try-catch blocks should be implemented to catch any errors that can't be explicitly coded around. Maintaining reliability will ensure an easier design experience for the future, and keep potential customers happy by having a device that works as much as it possibly can.

Software Maintainability

Software maintainability means to have a software design that can be worked upon and continued to be used even as a codebase grows. This is something many junior-level programmers struggle with as a lot of the time they are more focused on reaching a solution rather than finding the optimal solution. A very good example is designing the “Fizz Buzz” program. The idea of Fizz Buzz is to print out integers, and at every integer that is divisible by three, it will print “Fizz”, when divisible by five it will print “Buzz”, and when divisible by both it will print “FizzBuzz”. Many younger programmers that approach this problem will start a for loop, and attempt to use if-else statements using modulo arithmetic to catch the possible states of the program. However, if one does this, and is asked to expand upon it, it will become apparent that this implementation will add lines of code very rapidly, and the if-else statement for dividing by all will become nearly unreadable as the number of numbers grows. For the curious this problem is optimally solved by assigning variables and relating the words to the variables, that way one can just loop through, searching for the variables, and they can be expanded upon easily at a later time. Moving the focus away from the “FizzBuzz” example and back to the design, one needs to make sure the design is aiming at this style of maintainability and expandability. One needs to write the program and algorithm in such a way that they are writing the program to work correctly, but also to be worked upon by others in the future. Using variables with meaningful names, paying attention to runtime complexity of the algorithms, using proper spacing to provide readability to the code, and adding comments to give designers an idea about what each block of code does and how it behaves in the grand scheme of the entire program. Taking these steps as well as others will add to the ability to keep the software maintainable and allow others to expand upon the design in the future, as well as cut down design time that would be spent deciphering the current code.

Software Portability

Arguably the most challenging guideline to adhere to in this section, software portability is the idea that it will work the same in every environment. This can be difficult because there are so many different hardware and software combinations, it can be impossible to account for them all. Even with this design, the selections have been narrowed down to one of each category that focuses on ease of design. However, a designer needs to think about the possibilities for the future. Maybe the sponsor company will want to switch to the Raspberry Pi Zero W because it is a fraction of the cost of the Latte Panda. Maybe the design company will want to switch to their own custom Linux platform because they want more creative control over the high-level software. Maybe the design company will want to use a different driver because it is newer and more up-to-date. The sections in this paper talk a lot about future-proofing and the idea of adapting the design in the future, however, this applies to how the code is being written just as much as how the device itself is put together. This will allow designers to be able to adjust the device to suit the needs of the manufacturer, and still maintain usability with little to no changes to the code of the design.

Software Standards Conclusion

Many of the above guidelines blend the lines. A device needs to be secure to ensure the safety of the company and those involved with the product. The product needs to be reliable to ensure safety for those using it. A reliable device is most likely very maintainable, as it will have been designed well and offer new ideas to be implemented without much change. A portable device will need to be maintained, to ensure the smoothest transition possible when changing parts in a design. These are the core guidelines a software designer needs to understand and follow when making a design, and they will be present in the finished prototype of this design.

4.2 Related Constraints

In this section, we will present many of the constraints encountered while researching and building a prototype for this project. Like time, safety, environmental, and manufacturability constraints.

4.2.1 Time Constraints

One of the largest constraints we have on this project is time, and time management. It is critical for each member of the team to document their milestones in an organized and timely manner. If there is a missed milestone or time chapter our group project could severely suffer in terms of passing this class and graduating. Seeing as we did senior design 1 currently in the Spring, this means that we will be taking senior 2 in the Summer. You must have a working prototype by the end of senior design 2 to pass the class. The summer semester is shorter by a few weeks than a regular fall or spring semester. This means we must be on top of our milestones every week or we may run out of time. Moreover, we are doing this during a global pandemic where UCF is mostly closed, and all our team meetings are online. Even though we are a group we are mostly working individually out of our own homes. Online learning has also made it more difficult to talk with our professors and receive help. Shipping of products takes longer because most things are shipped to people's houses now and there aren’t as many people working out in the field.

4.2.2 Safety Constraints

There are many safety issues one can encounter when dealing with bad fuel. We know that the presence of water in fuel can lead to microbes growing and turning into sludge. This can rust and corrode whatever tank the fuel is in and it can even travel into gauges or the engine causing rust and damage there. This can lead to expensive fixes and possible replacement parts for the tank or engine. However, the worst that can happen is the possibility of the sludge completely blocking a part of the engine while it is on. That would be the worst-case scenario if you are in a plane and one of the engines goes out due to bad fuel. As seen and reported on we know that during emergencies like natural disasters is when we tend to use stored fuel if these fuel storage tanks aren’t checked regularly, they could produce bad fuel. From these reports, we know that the main reason for equipment failure in an emergency is due to bad fuel which is easily overlooked and something simple to check. It would be made simpler and faster by using this system.

The most important part of this project is to make sure it is a safe device to use. Considering we are using high-powered lasers and a very flammable sample we must take all the steps necessary to make sure no one gets hurt. To separate anything that can ignite away from where the sample is and constantly make sure the output power of the laser is not too high.

4.2.3 Environmental Constraints

The RSCL can also have a huge effect on the environment in terms of being able to test fuel quickly and not damage the sample. A stretch goal for this would be able to tell you how much water is in the fuel and determine if this is a dangerous amount or if there are any microbes in it already. This easy check would save many vehicles and engines not having parts thrown away due to corrosive fuel. It would also be helpful to see this before it went anywhere or was used for anything so if it is contaminated it can be disposed of properly in an environmentally friendly way. Although our world is moving towards electricity for power, I think fuel will always be an important backup. Especially now that we add so much to fuel to be able to stretch how much market fuel, we can get out of the raw fuel extracted, this makes the life span much shorter.

4.2.4 Manufacturability and Sustainability Constraints

The ability to manufacture this would be mostly restricted by the cost of parts and assembly available. Since we are using optics, lasers, and spectrometers the price for all these materials is around $40,000 not including the alignment procedure to make all this work. This would also be hard to do as there would need to be a casing to it protecting each of the elements from the others. As well as human error in case it is misused or mishandled. It is also important to consider that this device would need to be able to easily travel and go to different climates and terrains while still working the same. This would add more to the requirements needed for the quality of the product.

For sustainability, our testing would be limited because we would only be building one system. It would be important to consider most other climates and choose materials that would be much more resistant to and robust to withstand these different situations. A huge factor other than temperature, which is usually easy to measure is the humidity in the air. As we are measuring water content in the samples our system mustn't accidentally measure something it shouldn’t have. The way we can protect a lot of our elements is to put them into housings that will protect the smaller more delicate pieces easier.

5. System Design Details

In this section, we will discuss the different components that go into the RSCL device. As this is an optical system most of the parts used will be optical parts. However, we will touch base on software and hardware components.

5.1 Optical Design Details

Many of the physical components for this project are optical components. Using the correct parts for this system is crucial to make the whole system work and be accurate. An important part of picking all the optical components is knowing the specifications, size, and price. If a part is faulty or misaligns in the system, it is imperative to be able to diagnose exactly what went wrong and fix it quickly when it is in its final form. So, knowledge about each part of the system no matter how small is crucial.

5.1.1 Spectrometer

One of the most important and expensive pieces in this design is the spectrometer. We have been given a QEPro- Raman+ Spectrometer by Ocean Insight to use on this project. This spectrometer is already preconfigured for a wavelength of 785 nm for Raman application.

The QE Pro is named such that it is for quantum efficiency. Quantum efficiency is the responsivity, usually dealing with monochromatic light. It measures the amount of current that a photon will irradiate at a certain wavelength. It describes how many photons that hit the detector will create electrons. This is best for measuring Raman peaks as Raman is usually very low levels where we could compare this to a high-resolution spectrometer and see clearer peaks in the higher range.

The QE PRO spectrometer has a back-thinned FFT-CCD detector and gold-coated mirrors. These allow low noise and low fluorescence levels in the background when using a 785 nm laser. It is crucial to keep fluorescence out of a Raman system as the fluoresce can saturate the detector and spectrum without being able to see the Raman. Fluorescence the light excites the electrons at a higher energy state. A similar spectrometer is the Stellar Net Inc Raman spectrometer it is advertised as a good cost-efficient alternative spectrometer with similar results. We can compare this to the Hamamatsu Raman spectrometer. This spectrometer advertises that it is an ultra-compact miniature spectrometer with high efficiency. We can see a direct comparison of the specs against the QEPro and Raman-HR below in the table below. Having a good spectrometer is imperative to getting good data out of the setup especially with spectra as sensitive as Raman spectroscopy.

	QEPro-Raman+	Raman-HR-TEC-785	C13560
Brand	Ocean Insight	Stellar Net Inc	Hamamatsu
Dimensions (cm)	18.2 x 11.0 x 4.7	6 x 17 x 15.5	9.6 x 6.0 x 1.45
Dimensions (in)	7.17 x 4.33 x 1.85	2.4 x 6.7 x 6.1	3.8 x 2.4 x 0.6
Weight	1.15 kg (2.6 lbs.)	1.5 kg	90 g
Raman Shift	0 cm^{-1} – 3000 cm^{-1}	200 cm^{-1} – 2,750 cm^{-1}	400 cm^{-1} – 1850 cm^{-1}
Optical Resolution	0.72 nm (FWHM)	4 cm^{-1}	10 cm^{-1}
Integration Time	8 ms – 3600 s	NA	NA
Dynamic range	~85,000:1	NA	NA
Signal to Noise Ratio	1000:1 (single acquisition)	>1000:1	NA
Quantum Efficiency	90% (peak)	NA	NA
Entrance Slit	50 µm	NA	NA
Power Requirement	Supply voltage: 4.5 – 5.5 V	NA	Supply voltage: 5.25 V
Operating Temperature	0 °C – 50 °C	NA	15°C – 35 °C
Price	$15,000	$5,495.00	NA

Table 8. Comparing different spectrometers.

5.1.2 Lasers

We had a few options when it came to the light source. We know we needed two wavelengths, 680 nm, and 785 nm; they would also need to hit the same spot on the sample. The first idea of how to do this would be to get two separate lasers and combine them using a panda fiber. The other idea would be to build another optical system specifically for these two lasers to couple into the same fiber.

After much thought and contacting Innovative Photonic Solutions (IPS) they gave us the option of borrowing one of the dual-wavelength lasers that they use for a similar project. This allows us to simplify our optical system on the laser side. It is a single unit with two lasers coupled into a fiber adapter with an analog control to allow for each of the laser powers to be adjusted independently and easily switch between the wavelengths with a toggle switch. We are very thankful to have IPS sponsor us with this laser.

5.1.3 Beam Splitter and Mirrors

When choosing a beam splitter, it was important to think of the laser wavelengths involved as well as how the beam splitter operated. We want a beam splitter that can reflect both of our excitation wavelengths but allow the scattered light through. We chose to go with the Semrock Razor Edge. We can see in table 9 the specification differences between the Semrock filter and the Edmund Optics filter.

	785 - Razor Edge	**785 - Laser Line**
Brand	Semrock	Edmund Optics
Type	Long Pass Filter	Long Pass Filter
Dimensions	25 mm Diameter	25 mm Diameter
Blocking Wavelength	785 nm	785 nm
Angle of Incidence	0 ± 2 degrees	0 ± 2 degrees
Clear Aperture	≥ 22 mm	22 mm
Transmission Band	790.1 – 1770.7 nm	795.2 - 1770.7
Substrate	Fused Silica	Fused Silica
Transmission %	> 93	93
Edge Steepness	1.6 nm	3.9 nm
Transition Width	3.9 nm	<7.8 nm
Surface Quality	60-40	60-40
Mounted Thickness	3.5 mm	3.5 mm
Price	$1,025	$695

Table 9. Looking at the specification differences for each beam splitter.

5.1.4 Lenses

Like in most optical designs the classic way of manipulating the light is to use lenses, in this design it is no different. We have three lenses in total on the RSCL system, two of which are direct connects to the fibers and one is a free-standing lens that helps focus the beam into the sample. In order of the light path, we start with the collimating lens attached to the laser input fiber, we are using an off-the-shelf part from Thorlabs part number TC18FC-780. This lens is pre-aligned for the 780 nm wavelength and an 18 mm effective focal length using a triplet lens system with air in between the lenses, this provides even collimation of the beam. Still following the order of the beam path, after the light reflects off the beam splitter it goes towards the sample. For the best reading we want the light to be focused into the vile in the middle of the sample to do this we use an achromatic doublet with a focal length of 30 mm, this means the best focus is 30 mm away from the lens, so this is where you should position the sample. This is also our largest lens at an inch diameter, this is because we want to be able to collect as much collimated light as possible to focus onto the sample. We want this lens to be anti-reflective (AR) coated because we want to reduce as much refection as possible in the system where we don't want it. Continuing to follow the beam path, after the light hits the sample it goes back through the lens, which doesn't affect the beam, it transmits through the beam splitter, hits the mirrors that align it back into the fiber end for the spectrometer. This light has gone through many changes and filters that by the time it gets to the spectrometer it is a weak signal. To help with this we attach the F280FC-850, this lens is very similar to the lens attached to the laser end, however, it will focus higher level wavelengths into the fiber end.

5.1.5 Filters

Other than the laser the most important components in this design are filters. Filters as their name suggest are there to filter the light, currently on the system right now I have two filters to block out light below 785 nm as they would be used in a normal single wavelength Raman system. We are using these, for now, to prove that the system can work at least as a regular Raman system. However, with the second wavelength at 680 nm, we run into the problem with the light being reflected by the filters currently in the system. To fix this we either need to find 3 different filters or a custom-made filter that will allow both wavelengths to go through. The filters that we need would act as a clean-up filter for both the 680 nm and 785 nm wavelengths without cutting each other off. To visualize this, we can look at the figure below. We want to see the individual peaks for each wavelength with a sharp peak and no other noise between or around them. As previously stated, we could get a custom-made filter to do this. However, this does come at a very steep price and not many filter manufacturers would want to do make a single custom filter.

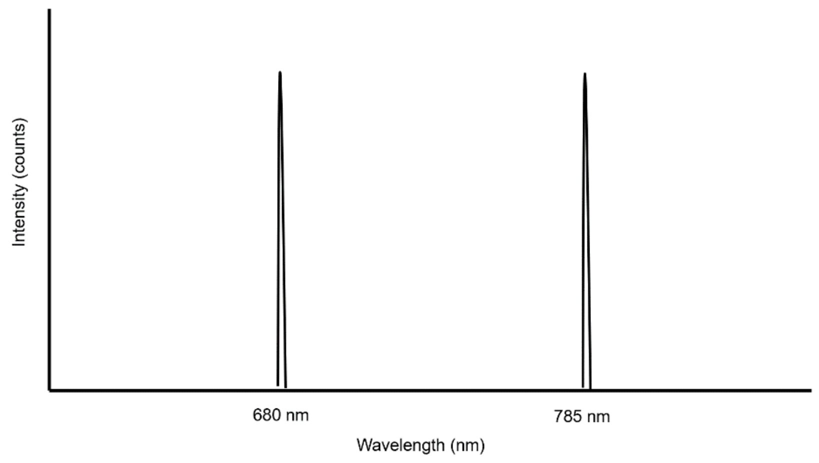


Figure 19. What we want our wavelengths to look like. Therefore, we need a cleanup filter.

The other option is to make a clean-up filter by using three different kinds of filters. These filters are easily visualized in Figure 20. Three filter system to make a custom clean up filter. We would need a short-pass filter (orange line) that would allow all wavelengths to pass up to the cut-off right after 680 nm, this is shown as 681 nm in the image. The next filter would be a bandpass filter (blue line) that would cut off everything up until right before 680 nm, as seen in the figure as 679 nm. This is acting as the first clean-up filter for the 680 nm wavelength. Continuing the bandpass filter would allow all light through until right after 785 nm, as seen in the image as 786 nm. The last filter is a long pass filter (green line) that would block light until right before the 785 nm wavelength, as seen in figure 784 nm.

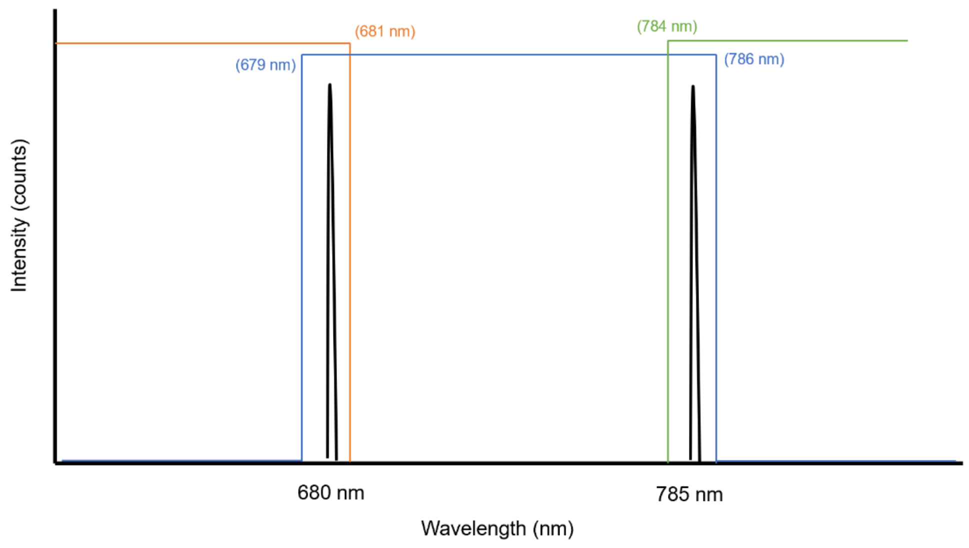


Figure 20. Three filter system to make a custom clean up filter

Currently, we are still in talks with Semrock and Edmund optics on whether to create a custom filter or attempt to find off a shelf filter to create this clean-up filter. Another option to this would be to find just the short pass filter that includes all wavelengths until right after 785 nm. This would help with making sure nothing above 785 nm enters the system but would still let the 680 nm through. However, it has proven to be difficult to even find a filter that does that. Many of the filters available look to do what we are looking for but the transmission cuts out at 775 nm. The reason we want to keep wavelengths especially over 785 nm at bay is that that is considered Raman signal but it is from the laser. This would travel through the system and show inaccurate data from the spectrometer. The long-pass filter at the end of the system plays a role in this because it accepts the wavelengths 785 nm and up. It will cut off anything below 785 anyway but it will still allow the laser, Raman.

5.2 Software Design Details

With the hardware and software components and elements selected, the design of the recipe algorithm, graphical user interface, and autonomous subroutine can now be described.

5.2.1 The Driver Software

To begin writing the recipe algorithm, one needs to consider how the computer (Latte Panda) will communicate with the spectrometer (Insert Spec Here). The computer and spectrometer are two separate components in the final design, and the spectrometer can't innately translate a light spectrum into a digital system for the computer to read.

Due to this, one will need driver software. A piece of software that converts the wavelengths and intensities captured by the lasers and spectrometer and converts it to useable digital data. Data that the design can further manipulate and present in a readable and informative fashion. There are three main driver software components to consider. Seabreeze a python-based driver, Omni-driver a java-based driver, and Ocean Direct a C++-based driver.

Seabreeze is a python-based driver that allows one to implement the data received from the spectrometer directly into a python script or program. This at face value sounds extremely beneficial because we have designated our primary programming language as Python. Seabreeze also has a decent amount of documentation and has instructions for use with Chocolatey and Anaconda. However, digging a bit deeper and with some referencing to the sponsor of this prototype design, Ocean Insight, one can learn that Seabreeze is outdated and sub-accurate compared to other drivers available.

Omni-Driver is another driver software that is made by the sponsor of this prototype design, Ocean Insight. It is a Java-based driver and allows direct support with MATLAB, is available compatible with all the sponsors' spectrometer products, and is available on all the major operating systems we mentioned in Section 3.6.1 (Operating System Selection). However, the fact that the driver is a java based does cause some conflict. Java isn't great at data manipulation, even at a base level. It also would require an extra runtime environment on the single board microcomputer, taking up more of the finite storage space that the design is aiming to have. Due to this, Omni-Driver most likely shouldn't be considered for this design.

Lastly, with both the analytical evidence above and a push from the prototype design's sponsor, Ocean Direct is the final driver to look at. This C++ driver is relatively new, surprisingly accurate in comparison to its competitors, and is particularly lightweight. The fact that it is C++-based means that we will not need another runtime environment on the board as the executable will be a .exe file. This saves us space and allows for a simpler software flow. One may be questioning that there may be problems in the design process since Ocean Direct is a C++-based driver and our main programming language is Python. This can be averted by the fact that we are using C++ solely for data collection and not anything else. Once the data is collected and exported, Python can completely take over for a seamless experience. More on this in Section 5.2.2 (Automation of The Software Design).

To conclude, Ocean Direct is the best driver software to consider in this case. Seabreeze is too outdated and inaccurate, and Omni-driver is too bulky to be of use. The lightweight design, despite the language conflict, indicates a clear winner.

5.2.2 Automation of The Software Design

While not a main requirement to have a working design, one feature that can help streamline the entire design, and add both flair and ease of use, is the process of automating the software portion of the design. The idea behind this portion of the design is to have the software execute immediately upon starting up the device. There should be no need for user interaction at any step of the software flow save for the portions of the graphical user interface (Section 5.3) where the user can select their options.

Since the Operating System for the prototype design has been chosen to be Windows 10, which can be found more in detail in the Operating System Selection section (Section 3.6.1), we can use a combination of .bat files and Visual Basic scripts to achieve this automation.

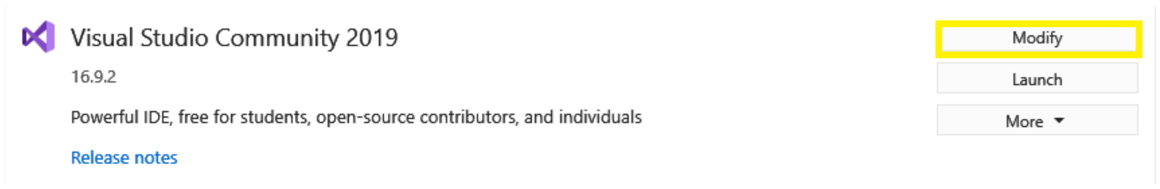
.bat (or “batch”) files are a native windows file type that contains a list of commands to be executed by the command-line interpreter. This can be used to run any amount of native windows commands, or commands that can be implemented. A detailed view of these commands with examples can be found in the appendix.

if the program “myprogram” is in the current working directory.

Visual Basic is an objected-oriented programming language with its integrated development environment. It can be used to create .vbs (or Visual Basic Scripts) which can interact with the .NET environment on Windows. Using these we can execute batch files with them or interact with the Windows environment from an entirely code-based approach.

Using a combination of these two, one can set up an entirely independent series of code executions to have a fully automated system. This document will now explain the setup of Visual Basic required, the commands needed, and the steps taken in the operating system to execute a simple python script immediately upon starting a local machine. This document will also include portions of the code to demonstrate the use of a C++ executable to show the use of Ocean Direct is not a conflict.

To begin, one needs to install Microsoft’s Visual Basic. Links to Visual Basic can be found in the Administration Section. However once installed, one will need to modify the Visual Basic to meet the needs to set up the automation. This can be done by searching for the Visual Studio installer in the windows search bar, opening it, and clicking modifies in the options.

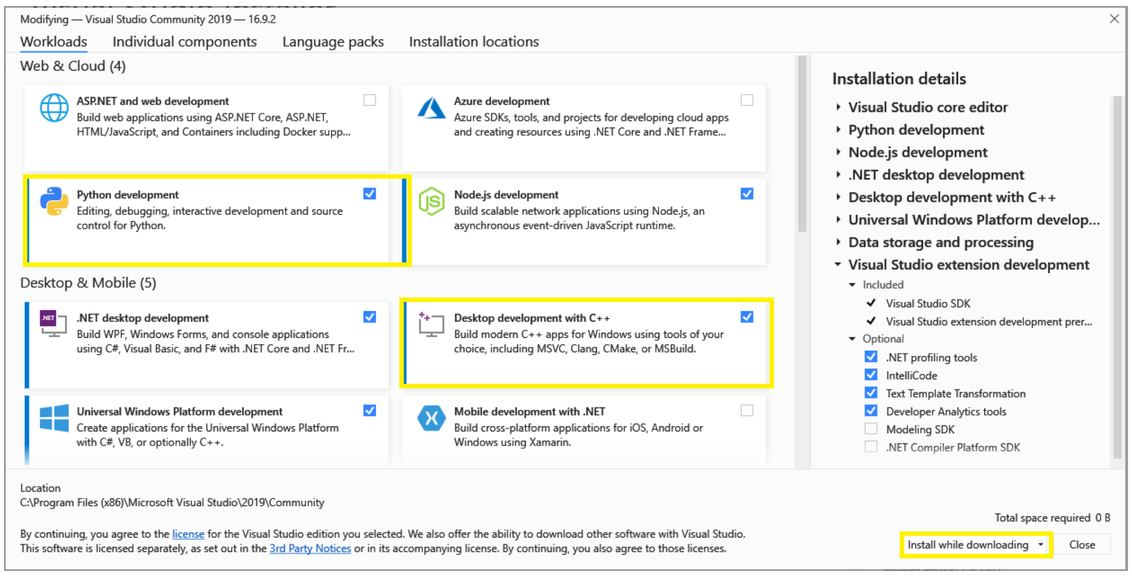

Visual Studio Community 2019

16.9.2

Powerful IDE, free for students, open-source contributors, and individuals

Release notes

Figure 21. The front page of the Microsoft Visual Studio program. The main option to click is modified to proceed with the tutorial.


Modifying — Visual Studio Community 2019 — 16.9.2

Web & Cloud (4)

- [ ] ASP.NET and web development — Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...
- [ ] Azure development — Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET Core and .NET Frame...
- [✓] Python development — Editing, debugging, interactive development and source control for Python.
- [✓] Node.js development — Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

- [✓] .NET desktop development — Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET Fr...
- [✓] Desktop development with C++ — Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.
- [✓] Universal Windows Platform development — Create applications for the Universal Windows Platform with C#, VB, or optionally C++.
- [ ] Mobile development with .NET — Build cross-platform applications for iOS, Android or Windows using Xamarin.

Installation details

- Visual Studio core editor
- Python development
- Node.js development
- .NET desktop development
- Desktop development with C++
- Universal Windows Platform develop...
- Data storage and processing
- Visual Studio extension development
  - Included
    - ✓ Visual Studio SDK
    - ✓ Visual Studio extension development prer...
  - Optional
    - [✓] .NET profiling tools
    - [✓] IntelliCode
    - [✓] Text Template Transformation
    - [✓] Developer Analytics tools
    - [ ] Modeling SDK
    - [ ] .NET Compiler Platform SDK

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 0 B

Figure 22. The main page of the Microsoft Visual Studio program. Here one can select and install their workloads to make the most out of the program for their project.

After proceeding with these steps, one will be presented with a window to modify and install workloads and packages within their own Visual Basic Studio, an example is displayed above. For our needs, one must select the “Python development” and “Desktop development with C++” workloads. Once selected the user can select to install the workloads which are displayed above.

Once installed, one can begin to write the scripts necessary to achieve an automated process. Starting with the batch file, one can create a blank text document and name it with a .bat extension. For this demonstration, test.bat will be used. Here one can write any commands they wish to be executed by the command-line compiler. Since the design is wanting to execute a Python script, we will add a line to execute a python script using a python compiler

It may be helpful to include the command @echo off at the beginning of the file to turn off on-screen text messages. If adding more commands, it may be helpful to add @pause to stop the batch file before executing commands after the pause command. Once stopped, the user can resume the batch file, in effect debugging the process.

Moving onto the Visual Basic script, one can create a new text file and name it with a .vbs extension. For this demonstration, test. VBS will be used. Here one can write any commands that they wish for the operating system to execute. It is this portion that will prove C++ has no conflict with the overall design, save for adding a few extra lines of code to the overall design. Using the command to create a shell script object we can control how the Visual Basic script will interact with the C++ program.

A shell script object will be executed by the Windows 10 shell, at the operating system level. Then one can assign our C++ program to a variable. For this demonstration, we will use exeName as our variable to store the string of the address of the test.exe file. Following this one can execute the program by adding a run command to our shell script object.

This command has three fields, the first is our variable name which is our C++ program. The second is an integer that represents if the command line window will be shown during execution. For Debugging purposes, "1" is selected to demonstrate the C++ program is executing. Finally, the third is a Boolean value, representing if the Visual Basic Script should wait for the C++ program to finish before moving onto the next line of code. Finally, we will add another shell script object for our batch file, this time for our batch file containing the commands to run our python script.

The last step needed to have a start to finish execution is adding our Visual Basic script to the registry. In the windows search bar, searching "Registry Editor" will pull up an application that allows one to edit their registry. It is important to take extreme caution when using the registry editor. Modifying the wrong register can lead to irreversible damage to one's operating system, with the only solution to have a complete re-install of the operating system, causing one to lose all their data on the system. In the registry editor, one can navigate to,

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

This directory contains all the registers related to startup processes. Right-clicking, one can select new -> String Value

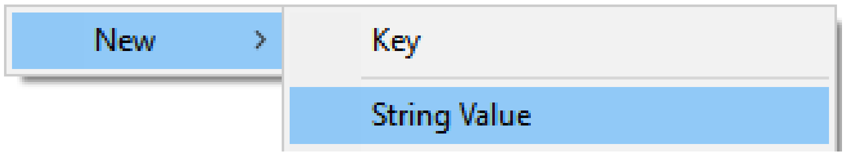


Figure 23. A screenshot of the actions taken to add a new String register to the registry editor. It is very important not to take any other actions besides those listed in this tutorial as it could lead to irreversible damage to one's operating system.

Clicking on the newly created variable, one can edit the value data to their Visual Basic script path. Again, it is imperative one uses extreme caution during these steps, as adding an invalid path to a real program or variable can lead to irreversible damage to the operating system. Once the register has been added, the automation setup has been completed. One can verify this by restarting their Windows local machine. After restarting the user should see the C++ program execute, and upon finishing the Python program or script should execute.

Some tools to help with the debugging process in setting up this automation could be a few handy Python libraries. The OS library will be necessary to switch our working directory when executing the batch file. This is because the working directory of the batch file is the start-up directory, so any files we want to work with inside of our Python program or script will not be present. This can be solved with the line, by changing the working directory with the OS libraries os. dir command, which will switch the working directory to the path indicated.

Another tool is a library called toast, which can be downloaded in the Windows command line. Toast allows the user to enable windows notifications from their Python programs or scripts. Notifications are found on the right side of the Windows 10 desktop. The command used for debugging in the prior demonstration showed. The following command shows a notification upon reaching this line in the execution process. The three fields are the title of the notification, the message itself, and the duration in seconds.

While the above demonstration was a brief proof that the automation can be done, one can see that with some small adjustments, the entire software process of the prototype design can be executed automatically with no user interaction. This can lead to a much more pleasant user experience and take time off the overall execution time of the entire machine.

5.2.3 The Recipe Design

In this section, the document will discuss the main bulk and most important piece of the software design process, the recipe algorithm. One must be familiar with the above sections in the software design process, as they will be referenced heavily throughout this description.

With the necessary operating system installed, and in that operating system, one has the correct python libraries, Visual Basic Studio and Workloads, spectrometer drivers, and compilers, one will have everything they need to create the recipe algorithm from scratch. To be clear, this process does not describe the interactions with the graphical user interface and describes very little the interactions with the autonomous subroutine system. The focus of this section is determining the requirements that the recipe design will deliver, and how it will do so in the software logic.

To begin, one can create a blank C++(.cpp) program file. In this file, we will use the driver commands to find the spectrometer on the computer. This assumes that the spectrometer is connected correctly and receiving power from a wall outlet or other power source. With the spectrometer located we can store it as a variable to reference later. The program can now set an integration time which will be determined by the user in the graphical user interface. Then, the command to read the sample will be executed. This will create two arrays for wavelengths and intensities respectively. With this data collected one can export the data as a spreadsheet file, for debugging purposes comma-separated values (.csv) file types are generally used. This information will then be passed to the python program, where the main logic for the recipe algorithm is located. The program should read the spreadsheet file and store the columns in NumPy arrays for further data manipulation.

With the NumPy array successfully made from the data pulled by the spectrometer, we can begin to implement the math used in Raman Spectroscopy. The goal of the algorithm is to be able to detect any major peaks that the spectrometer managed to pick up. However, there is a possible side effect that can occur during the process of taking a reading. We can experience a peak that is before, and larger than the peak that we are looking for to use the capabilities of Raman Spectroscopy. This raises the question of how to handle this event? How do we tell the computer to ignore this peak, where the entire goal of the algorithm is to find the largest peak? The answer requires the foreknowledge that this initial peak will match, to a certain extent, the intensity of the laser, which will be a set value by default. We can then write the algorithm in such a way that it ignores values that reach anywhere near this inputted value.

With this done, the algorithm can then begin to search for any peaks, highlighting those deemed necessary, which will be determined by the mathematics for Raman Spectroscopy. This information will then be sent to the graphical user interface, to inform the user of the information. This is usually meant to be purely informative for the user and not deterministic. The deterministic aspect of the algorithm is to be able to detect a chemical substance other than the fuel inside the sample. It is important to note that the initial goal of this prototype design is to detect contamination of any sort, with a stretch goal being to detect water contamination, followed by a quantity of the water contamination, and then the ability to detect other types of contaminating chemicals.

While this all is certainly possible, it is uncertain if it will be accomplished in the time frame for the prototype design. Assuming this is however possible, one could be able to train the algorithm to detect water by first identifying the value of the peak that represents water in the sample. This could then be compared to the collected sample to determine the contamination in the fuel sample. To go further with this idea, based on the difference between the two values, it can become possible to be able to determine how much contamination is in the collected sample. With this logic implemented into the algorithm, one can be able to repeat the process with other common chemicals that can be found inside of fuel, adding to the idea that the final design will be an all-encompassing fuel testing tool. This however will require the gathering of data, and even with this, it will most likely not be a perfect algorithm, as many factors can influence the values obtained from a source sample as well as a test sample.

It is here that the idea of implementing some sort of Machine Learning process comes into play. Machine Learning can take the combination of many samples and determine the similarities and differences in them. This could be set up to analyze the sample of water, acetone, glucose, or virtually any chemical. Then the samples are added to an overall dataset, which runs through the Machine Learning algorithm. It is important to note that this would require a supervised learning style of Machine Learning, where a user gives the proper dataset that represents a correct outcome and allows the algorithm to determine if a separate dataset is related at all. This would be like saying to the algorithm, “Here is a sample with water in it. Is this different sample like water at all?” This is different from unsupervised learning, which would take a set of multiple different samples and group them based on differences. While this style of learning is interesting and can yield useful information, it will most likely be better to train the algorithm chemical by chemical. All of this in the end will yield a more accurate detection algorithm to determine what types of chemicals are inside of a user’s fuel sample in the field.

With this information collected, another stretch goal that has been determined is to design an algorithm to recommend a treatment for the contaminated fuel sample. This can be done by hard coding the types of treatments required for each chemical. Much like treating a residential pool, each fuel sample can require a certain amount of treatment to be kept up to or meet a certain standard. This can be done in the same manner as the detection algorithm which will allow designers to determine the required treatments for each chemical. Likewise, we can also use the same Machine Learning Process to improve upon the recipe treatment algorithm, we can present the algorithm with a contaminated substance, allow it to determine the type of contamination, and then have it recommend a treatment that a designer can approve. This will lead to better treatment algorithms as well as saving money for the user as the number of chemicals needed to treat substances will be more accurate, leading to less waste.

5.3 The Graphical User Interface

With the recipe algorithm giving accurate data from the spectrometer, there needs to be a way of presenting this data concisely, but also informative enough for an average user in the field to obtain the necessary knowledge they need to continue their job function. The info market test kits provide also needs to be considered, as the final product is intended to beat out those kits at most tasks and angles. To achieve this, the design will need a graphical user interface. The graphical user interface will not only be used to display the data on a screen for the user's convenience, but it will also utilize the screen's touch screen capabilities for the user to interact with and start the machine and the corresponding algorithm. It is important to note that the graphical user interface would be designed on the surface by a professional graphic designer, whether that be talent hired or already existing within the product designer's company or outsourced out to a professional graphic design company. This prototype will use a rough draft design, to demonstrate the interactivity that a user can have with the design. This can be transformed into a professional design with little to no difficulty later in the design cycle.

5.3.1 Choosing a Graphic Design Software

While the design is meant only to be a rough draft that will be improved upon in later stages in the design process (closer to a final product), the rough draft still needs to be flexible and convenient enough to be able to interact with the code to make an autonomous and user-friendly experience, both for the designer and the intended customer or user. To achieve this the design software used to create the graphical user interface needs to be carefully considered. A blend between capability, ease of use, and scalability need to be considered. The design needs to be able to be taken apart piece by piece to be able to be represented and interacted with by the code.

One of the most popular graphic design software tools is GIMP. GIMP is an open-source, cross-platform image manipulation program. It is free software that allows users with the required knowledge to modify it to their needs. The software offers a wide variety of 3rd party plugins that can enhance the user experience. It has a decent amount of documentation and support from 3rd parties. Unfortunately, where GIMP falls short is its layout. The layout is not user-friendly in any way, requiring previous knowledge of graphical design tools to understand. Alongside this, GIMP does not handle creating images and designs from scratch. It excels in editing and manipulating graphical designs, however, to create something new and unique, one must look elsewhere.

My Paint is another graphic design software tool that is relatively commonly used. It is also free and open-source, offering an interesting endless canvas and a quality brush engine. It is available on the three popular platforms (Linux, macOS, Windows) and offers seamless transition between the three. The only major drawback is the lack of features. The program is mainly intended to just create images/designs and not to edit them. If one were to want to go back to retouch their design, it may be more beneficial to look elsewhere.

Lastly, Krita should be considered. Krita is another free, open-source graphic design software. However, Krita takes the pros of GIMP and My Paint and does a decent job at covering up their cons. Krita offers a large amount of documentation and resources, including free courses to help one learn the tool and become a better artist. It has a very user-friendly user interface, allowing those with limited graphic design knowledge to easily get started in creating a design, it has a very strong brush engine, allowing users to make use of different types of brushes, stabilizers, and vectors, and even offers interaction with PyQT, a custom API designed by Krita to interact with Python Scripting.

Needless to say, Krita will be the software tool used to design a rough draft of the graphical user interface, as it allows one to easily start a design, and has potential future-proofing features in the form of the Python scripting interaction.

5.3.2 Graphical User Interface Design Concept

With the graphic design software chosen, the graphical user interface can begin to take form. To make the transition from concept to code as seamless as possible, each portion of the design needs to be its independent layer, which can be exported to its image later. It is important to note that this is a rough concept, and even at the end of the prototype design stage features may change including additions and eliminations. With that in mind, one can now begin analyzing the potential features in the rough draft of the design.

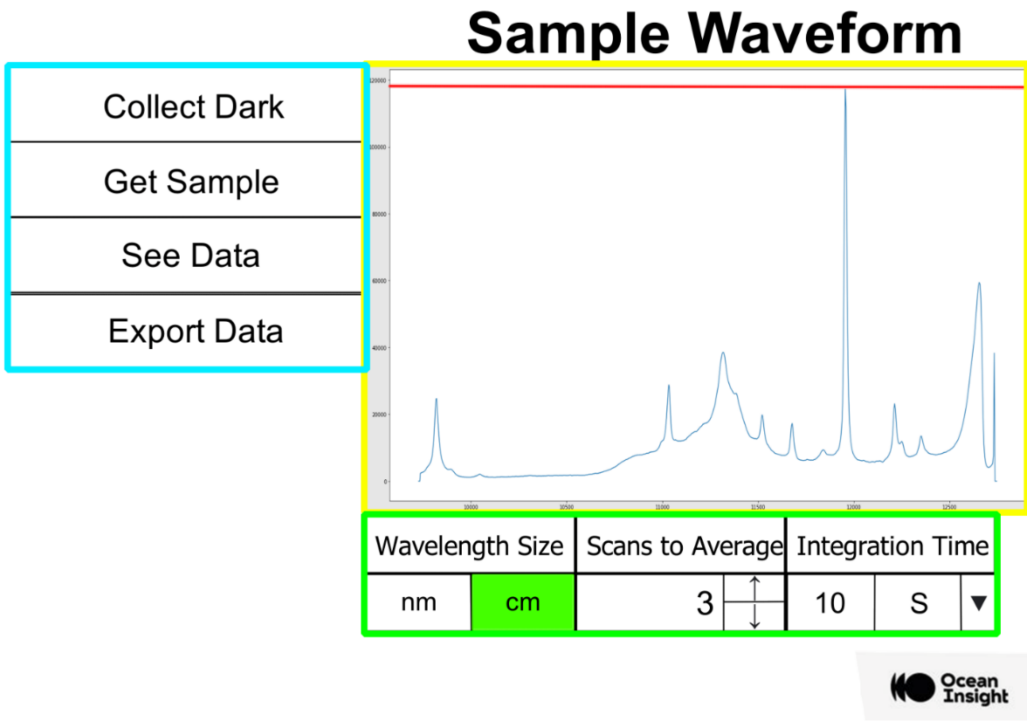


Figure 24. The first draft of the graphical user interface. The three main areas are the Yellow box which highlights the sample waveform, the blue box which highlights the powers options for the user, and the green box which highlights waveform options. This design is subject to change on the delivery of the final prototype.

The first feature to look at will be the waveform graph, highlighted in yellow. This graph displays the waveform that was translated from the optics to an analog waveform by the spectrometer. Using this data, the recipe algorithm will determine the chemicals in the tested sample. Missing from this design are axis markings, indicating the wavelengths in nanometers and the intensities. This hard information could be important to a user for calibration, as well as seeing any additional waves that could have a meaning which the recipe algorithm failed to detect. Along with the graph are reference lines that will indicate the peaks and valleys depending on the methods used to collect information.

The next feature to note is the options, highlighted in blue. These will be the main interactive points for the users. One will mainly look at the three main items which are Get Sample, See Data, and Export data. The Get Sample item will allow the users to run the device and collect the waveform seen on the right. This is the main interaction of the device and interacts with most parts. Next will be the See Data option, which will replace the waveform with a spreadsheet-style table of the data collected. There is potential for highly certain numerical values like peaks and valleys that have a particular significance. Lastly, Export Data will send the data to a .csv spread, with the potential to allow the user to expand the selection to other file types (i.e. .xlsx) or send the data directly to a SQL database for further queries and data manipulation which would have a hand in the machine learning potential talked about earlier in this document.

The last set of features to look at are the waveform options highlighted in green. These options will allow more dynamic control over the data that is collected, displayed, and possibly stored. The first option is the wavelength size. One can switch between nanometers and centimeters, which will be determined by taping the desired option causing the background to change to green. The default will be nanometers because this is the most common setting in the industry. The next option is the scans to the average section. Here one can determine the average number of scans they wish the spectrometer to capture. Lastly, the integration time section, which will have an integer value set by the user, and a drop-down menu to select between seconds, milliseconds, and microseconds.

To conclude this section, the graphical user interface succeeds in giving a rough concept of how the user would interact with the device. The device would autonomously power up into this view and wait for user interaction. The use of Krita would allow the design to be translated easily into the software space and allow for future scripting interactions with its very own Python API.

5.3.3 Coding the Graphical User Interface

As stated in Section 3.6.2 we will be programming the graphical user interface in Python. This is done to keep consistency and the number of languages used in the prototype design to an absolute minimum. Java would most likely be easier to code a graphical user interface since it has more support and documentation as well as many libraries that help with graphical user interfaces, but we want to avoid Java for the most part in this project as it is becoming outdated and can be cumbersome and clunky. Python still has many tools to help with graphical user interfaces, so coding one shouldn’t be too much more of a challenge than coding one in Java.

To code up a graphical user interface in Python, we will need some tools to achieve this task. One of these tools is a Python package called Tkinter. Tkinter is a Python interface tool kit that is available on Windows systems and most Unix platforms. Tkinter itself is not a part of Python natively but can be installed in the same manner as installing toast in Section 5.2.2. Tkinter has a moderate amount of documentation and support as well as some key modules that will aid in the challenge of designing this graphical user interface.

The main concept of the design is to have an object that we will call the root. The root will be the main object that we attach everything to. Whether that be a window, a frame, an image, a button, it all will be attached to the root. This can be somewhat conceptualized as a more complex linked list or binary tree. We have many different objects that all stem from a singular object. Adding the Tk method to this root will allow us to open a window in our operating system with our Python program. Initially, this window will be blank, however, this will be added as the graphical user interface is fleshed out.

One type of object that could be attached to the root is the canvas object. The Tkinter canvas object can be used to draw in the window that we created. One can draw graphs, plots, or even add widgets like images and rectangles. The canvas object uses a coordinate system to orient where each object will go, with x = 0 and y = 0 being the top left corner. The canvas object also can determine the window size, which will be in pixels, by using the height and width variables. Canvas also has a background variable that will change the color of our background. It is important to note that to make your graphical user interface appear with the properties specified in the canvas object, that you call the pack method on canvas. This will render your canvas method in your root object. Without it, the root object will just display a blank window as if you just had the empty root object.

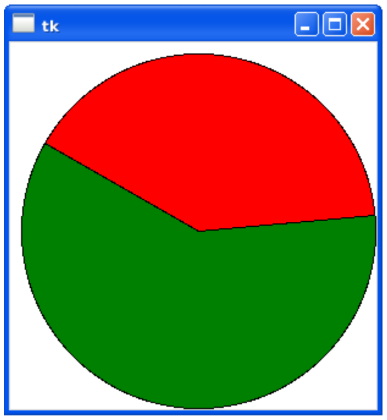


Figure 25. An example of a canvas with a pie chart and a white background. One can begin to see what sorts of widgets could be placed inside of this window, such as images, other types of graphs, and charts.

Another important object to use will be the Frame object. The Tkinter frame allows the user to organize and group widgets into a container in which they can be placed. One will likely be using many widgets in designing a graphical user interface, so it is important to be able to organize those widgets and be able to manage them one by one. Frames allow one to add text, justify the objects inside of the frame to either side and offer padding for more precise placement of the objects.

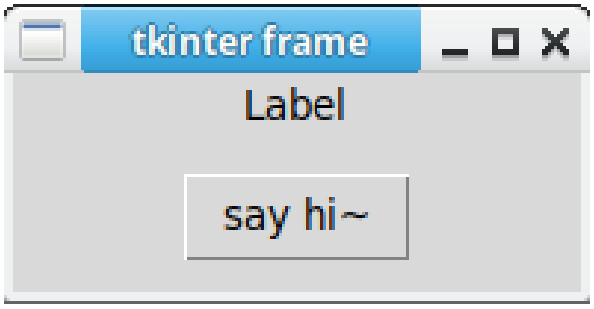


Figure 26. A simple example of a frame in Tkinter with a label and a text of "say hi~" One can imagine how you can expand upon this simple concept.

Looking at the rough draft of the graphical user interface (Figure 24), something that will need to be added is buttons. Tkinter also has a button widget that can be attached to your root or your frame. These objects can display text or images that convey purpose to the user. One can attach a method to these buttons to perform an action when the user interacts with the button. This is done with the command method. The button object also has background and foreground color controls, as well as the size and padding variables. The button object also contains text control variables such as justify, position, underline, and wrap length. One will also need to call the pack method at the end of the button object to render the button in our root window.

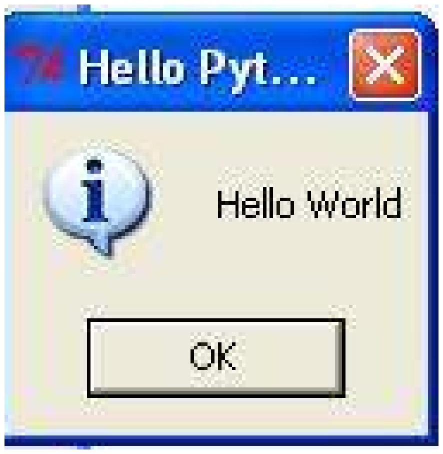


Figure 27. An example of a button object being used in a frame of a root window. One can distinguish between the three objects by noting that the root is the window itself, the frame is the beige color, and the button is the rectangle with the text "OK".

Another useful tool in Tkinter is the file dialog module. This module allows one to be able to manage and use files inside of the graphical user interface. The file dialog module has the parent object which specifies the window to place the dialog box on top of. One can also use the initial file and intialdir methods to aid in navigating the user's directories or to limit the user to certain directories and file types. One can also use the default extension method to give a default extension type to any files one wants to save with the inherently native save dialogs. This module will be useful for the prototype design's graphical user interface for the "Save Data" option. One can begin to imagine the steps to take to design this interaction by putting together the frame, button, and save dialog methods. The button inside of the mainframe will be clicked, with the button's command method executing and calling the file dialog module to save the collected data from the spectrometer.

Another simple Tkinter object that will be used in the graphical user interface is the label object. The label object will allow us to add titles and text-independent of other objects, aside from parent objects like the root, canvas, or frame. This will also add more control to these titles by giving us the ability to control the anchor point, the background color, add borders, change fonts change cursor types, control text justification, add padding, and adjust wrap length. This all can be done with text or images.

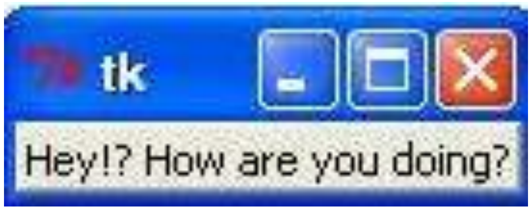


Figure 28. An example of a label inside of a canvas. This label would be independent of all other objects if any others were present.

A possible interaction to consider when designing the graphical user interface is to use the OS library that was mentioned in Section 5.2.2, this is because the OS library has a command called start file which will allow one to start an executable from the graphical user interface itself. Since our C++ program will have no physical interaction, and will just be the spectrometer collecting data, we don't need the C++ program to run "independently" from our python program inside of our Visual Basic Script. The start file command can run the C++ program while the Python program waits for it to finish, and then since the Python program is not interacting directly with the C++ program, and just the retrieved comma-separated values file, the need to interact with the program inside of the Visual Basic script is not needed. This can reduce code complexity and size, as well as reduce the overall encumbrance of needing to use two programming languages.

With the tools above, one can begin to imagine how the rough draft of the graphical user interface can be pieced together. Each object in the physical design of the graphical user interface needs to be its image, as we will need to call these images for each interaction. For example, the get sample option in the rough draft will need to be its image because it will be its button. We can’t have them mixing with the other buttons, as this would just create one big button with a single functionality. However, these problems were considered when choosing the graphical design suite. With all this in mind, the graphical user interface begins to come together, and one could begin to think about how to create their graphical user interface.

5.3.4 Overview of the Software Flow

To conclude the software design process chapter, we will give a general overview of the software flow. This can be summarized in Figure 4; however, more informative detail will be used here.

The software end of the design begins with the device being powered on. It is at this point the Operating Systems goes through its start-up procedures, which we have modified by editing the registry in Section 5.2.2. This will cause the Operating System to automatically begin the Visual Basic software, which in turn will start the Python program containing the graphical user interface. It is at this point the device will remain idle until the user has inputted their desired settings and executed instruction by selecting a command from the graphical user interface. We will assume for the sake of the example that the user has asked for a sample to be collected as this option will have the most interactions with it.

The device will then resume the Python program, which will send a signal to the Visual Basic program to start the collection process. This will execute the C++ program, which will start the spectrometer and collect data, turning the data into readable values, and saving them to a CSV file. This will be treated as the excel file in Figure 4. The C++ program ends on its own and the Visual Basic script will return control to the Python program, which now has a usable data file. It will import the data file, turning the values into a NumPy array for easy handling.

Next, the program will search through the values, to find peaks that are relevant to the Raman Spectroscopy mathematics to highlight and obtain useful information. With this foundation, the program will display the graph and the highlighted or otherwise indicated peaks, as well as begin to determine if the sample is contaminated.

If the stretch goals have been implemented, the program will also determine the type of contamination, sending this information to the graphical user interface, and then begins to do the calculations for treatment. Once the treatment has been determined, this information will be displayed to the user. Note that the graphical user interface may become cluttered with all the information listed above. It is currently undecided if the above information will be displayed on a single screen, however, if it proves to be too much information at once, the treatment values can be saved to a text file, and an option to “View Treatment” can be added to the graphical user interface to display the calculated values. This would just involve writing a portion of code that fetches and displays the text document, which is trivial and adds next to no space to the code.

Once this has all been finished, the program will then wait for the user to input another command, or to initiate a power-off sequence. Note that the current rough draft of the UI does not include the power-off option, though it could be easily modified to do so. This signal would tell the Python program to quit while raising a flag to the Visual Basic program which will tell the Operating System to shut the device down.

With this, the device has completed the software flow. This would be modified depending on the command given at the UI stage, however, one can easily see of imagining the type of adjustments that would take place, as none of them seem to be any more complicated than the sample collection process.

5.4 Possible Future Software Implementations

In this section, the document will discuss possible implementations and improvements in the software space. These implementations will not be included with the prototype software design, but the prototype software design is designed in such a way that it is open-ended and can be modified and improved upon. This is done so as it is the industry standard to do so, and more than often leads to a better overall design in general.

The first possible implementation would be the introduction of a machine learning component to the design. This would mainly target the recipe algorithm portion of the design. The main goal of this would be to improve upon the accuracy of the recommended treatment of a given sample. Therefore, the export data feature exists, and why the ability to determine a file type for the exported data has been implemented instead of just forcing the file types to be all comma-separated values file types. The increased accuracy could lead to a more attractive product or could be a driving price point factor.

The main drawback to machine learning is that it can be very time-consuming and by extension expensive. One of the major problems that software engineers working with machine learning techniques experience is that finding good, clean, and informative data can be difficult. Data often has holes and discrepancies. These issues force engineers to have to analyze data by hand, and either correct the data, work around the issues, or remove the data sample from the algorithm.

One way this could be supplemented is to require users to send the data to the design company. This would provide a constant supply of data (if the product sells well) and can be implemented without the need for user intervention by creating socket programs to immediately send the data to servers over an internet connection.

This leads to the next implementation which is almost an extension of the machine learning implementation, creating a database. One of the major sections in the field of software engineering is cloud data computing. Creating a SQL server/database would allow easy and smart storage of the collected data, whether this is the data automatically sent to the design company, or an entirely separate database for the user's company to store the data of their client/tested samples. This data can then be analyzed by data scientists, and used to create informative dashboards by statisticians to give a more informative view of the data that can't be implemented in the final design of the prototype device, or would overcomplicate the view of the design, conflicting with the design specification that the view should be relatively simple so a user needs minimal knowledge of the device and field.

Another possible implementation would be to convert the design to a single language. In this prototype, the design uses three languages, Python, C++, and Visual Basic, to achieve the expected results. However, while this isn't a problem for the prototype design, this does have the drawback of not streamlining and adding an inflated code size to the overall design. The code size would be an issue if the mainboard for the computer switched to a microcontroller of some kind, or a smaller scale board like a Raspberry Pi. The streamline portion would just be for conciseness and to allow the addition of future designers to have an easier time adjusting and understanding the project. This however would require the main algorithm and graphical user interface to be written in C++ or to create a new spectrometer driver written in Python. However, this would allow the use of smaller scaled boards, and possibly increase the sample collection and processing speed.

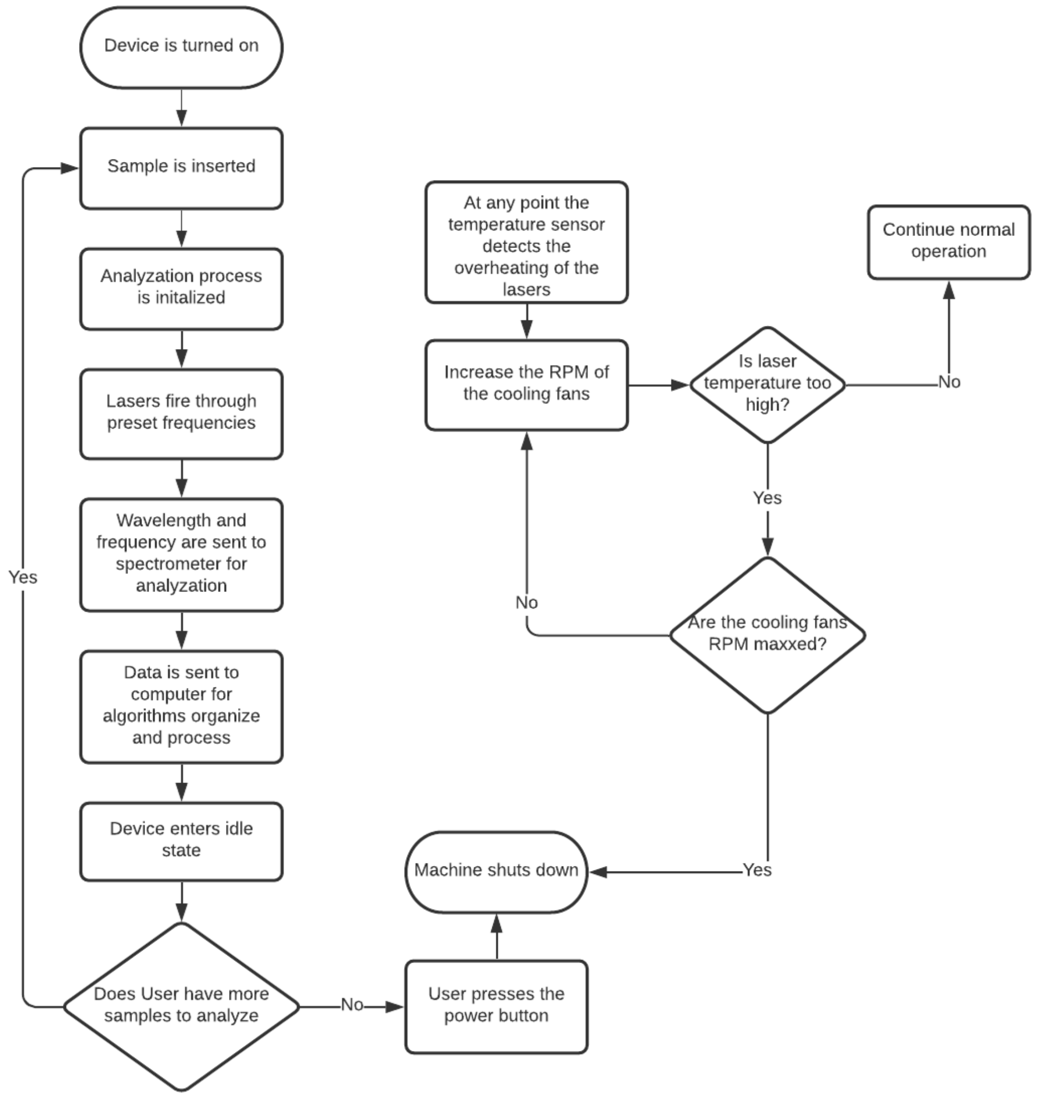
For the graphical user interface, future implementations would most likely be having a professional graphic design company design the layered images. While the graphical user interface is primarily code-based, the images are what the user experiences, and so having quality images will allow users to be less overwhelmed by the clunky feeling rough draft design. The graphical user interface will also be updated with more options depending on the future implementations to the design, to allow the user to interact more with them.

The autonomous subroutine will be updated with the programming languages become streamlined as described above. The Visual Basic component will remain but will be reduced to just starting a particular program and letting it run until the user decides to power off the device. This is however assuming that the main operating system remains as Windows 10, or another Windows version. Switching to Linux or macOS will require a complete rewrite most likely, which will require more testing and development.

Finally, the recipe algorithm will most likely benefit from the addition of more chemicals and substances. This is where most of the above implementations could play a role. Adding more chemicals and substances would allow the device to be more versatile and add more to the possible recipe treatment plan. However, this would require more testing to be able to verify the presence of said chemicals, and more data collection if the machine learning component was implemented. In the end, though this would make the product more attractive to potential buyers, and further the end goal of designing a product that can eliminate current market test kits from the competition.

5.5 Operation Flowchart

Here is the flowchart for the operation of our prototype, where it demonstrates the processes that the prototype will go through to deliver the result. As well as multiple conditional statements depending on what happens during its operation.



Device is turned on
Sample is inserted
Analyzation process is initalized
Lasers fire through preset frequencies
Wavelength and frequency are sent to spectrometer for analyzation
Data is sent to computer for algorithms organize and process
Device enters idle state
Does User have more samples to analyze
Yes
No
User presses the power button
Machine shuts down
At any point the temperature sensor detects the overheating of the lasers
Increase the RPM of the cooling fans
Is laser temperature too high?
No
Continue normal operation
Yes
Are the cooling fans RPM maxxed?
No
Yes

Figure 29. Operation Flowchart

5.6 Mechanical Design Details

For the mechanical or hardware side of the optics, we decided to go with a cage system. This is because it is easy to construct while keeping all the optics aligned. This is very helpful as prototypes or experiments that involve optics are usually done on an optical table or breadboard. Breadboards are heavy and often need more mechanical components to hold the different optics making the system bigger than it needs to be. The other side-effect to using a breadboard is that the components needed to hold the optics are attached to posts that go into post holders that attach to the table or board, this creates a lot of variables in the system that would take longer to be aligned. For these reasons, we decided to go with a cage system. This makes it possible, so the optical system is easily portable and adjustable in the x-direction for the lenses and filters. The mirrors are in right angle kinematic mirror mounts that allow for tip-tilt adjustment for the light to be aligned into the fiber.

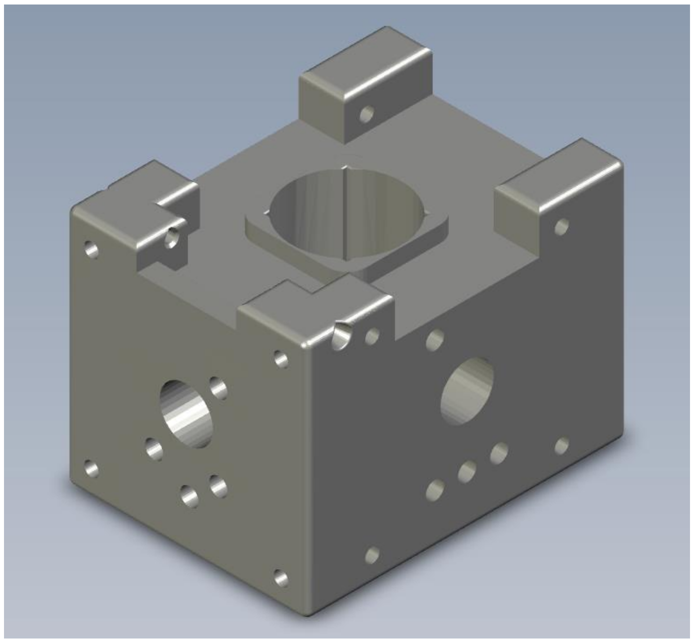


Figure 30. Side view of the vial holder, as we can see the light will come through a hole on the side, and the vial is dropped into the top.

The cuvette/vial holder is a 3D-printed part that easily screws onto the end of the rods for the cage system allowing for a smooth alignment process. As seen in figure 23 the print comes with holes on each side to access the cuvette. These can either be covered by black tape or have mirrors placed in them for a higher scatter reading. For this application, we will not be adding mirrors to the vial holder.

Using a vial instead of cuvettes was decided as it is more cost-efficient to use vials seeing as they usually come in glass and with the chemical samples we will be using it wouldn't be easy to reuse the same vial for different samples. Cuvettes would be nice as they are flat and square on the surfaces and many may think this would be better for optical readings. However, for the application we are doing the roundness of the vial doesn't matter as we can just focus the beam to the center of the vial where the sample is rather than the surface.

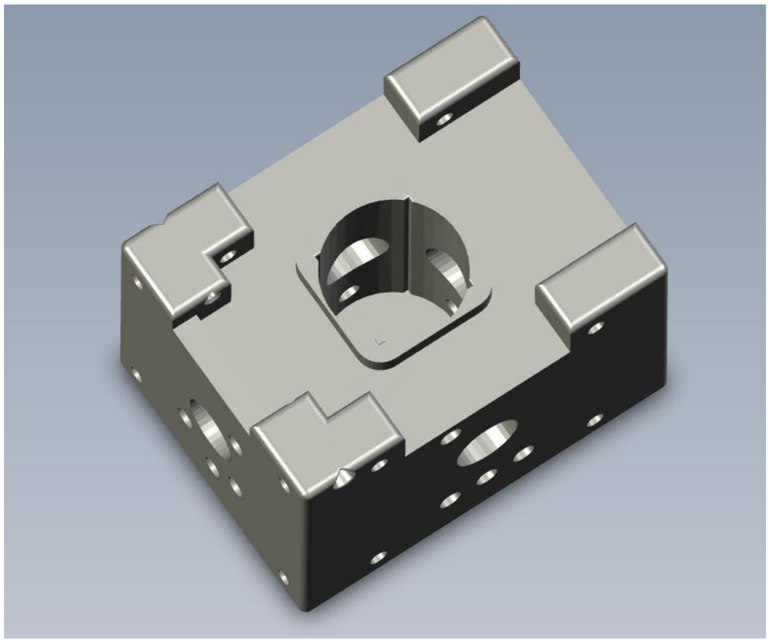


Figure 31. Top view of the vial holder. We can see how all the holes connect to the middle of the vial. We will be blocking 3 of these side holes.

Here we will be looking to implement a box around the optical cage to keep all the optical components together while also keeping all the laser light contained and safe for people to use. The goal is that this box would be able to have all the components except the laser in it, including the spectrometer, the computer interface, and the screen in hopes to make this system portable. The box will only allow the cuvette hole to be accessible by the user, making it so it would be very simple to take a reading without having optics knowledge, as well as keeping the system aligned without environmental factors taking effect on the optics. This would also come with a cap or cover for the cuvette holder where if the cap is open the laser cannot turn on. When the cap is closed the laser can turn on and this will act as a complete dark for the sample readings.

5.6.1 Mirror mount adjustment knob shields

To make the system easily portable especially without an outer case to protect the caged optical system, we created 3-d printed covers to fit over the mirror adjustment knobs to avoid misalignment when traveling. Considering everything else on the cage system is screwed down with these as the only variables it is very convenient to screw the covers on by the top screw to make the system portable. This came in handy when demonstrating the optical cage system worked to the CREOL staff for this senior design project. We can see images of the covers at different angles in the figure below.

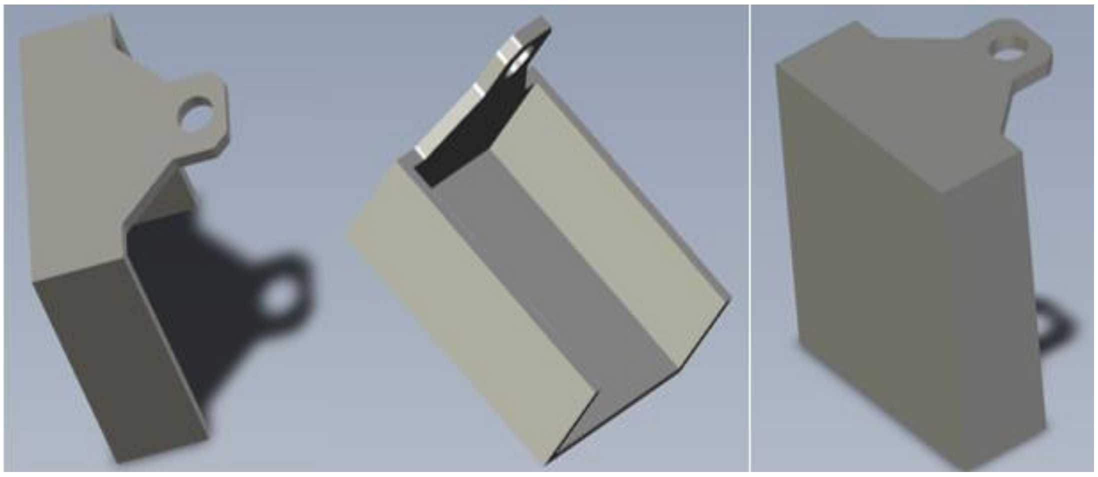


Figure 32. Mirror mount adjustment knob cover

For a better understanding, below is an image of the two kinematic mirror mounts that would be covered by the shields. As we can see they are easily reachable and adjustable this is great for aligning. However, for packaging and moving the system they can easily be knocked around causing misalignment.

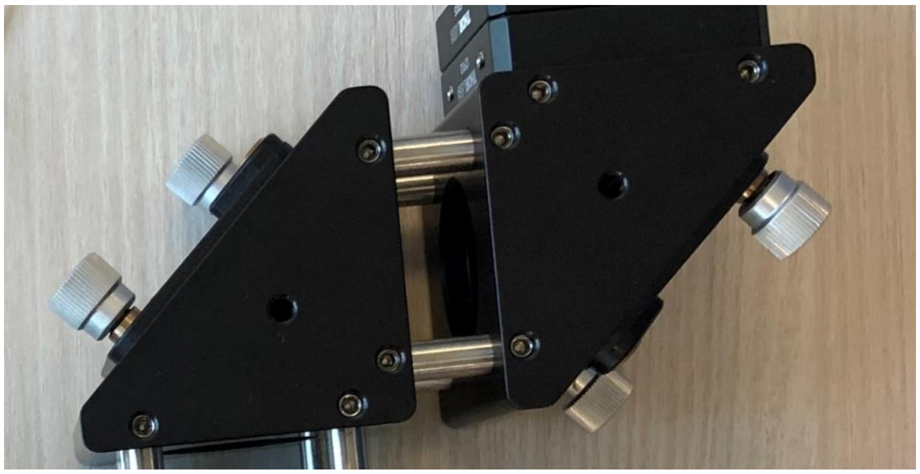


Figure 33. Kinematic Mirror Mounts

5.6.2 Optical Box Hardware Design

Here we will be looking to implement a box around the optical cage to keep all the optical components together while also keeping all the laser light contained and safe for people to use. The goal is that this box would be able to have all the components except the laser in it, including the spectrometer, the computer interface, and the screen in hopes to make this system portable. The box will only allow the cuvette hole to be accessible by the user, making it so it would be very simple to take a reading without having optics knowledge, as well as keeping the system aligned without environmental factors taking effect on the optics. This would also come with a cap or cover for the cuvette holder where if the cap is open the laser cannot turn on. When the cap is closed the laser can turn on and this will act as a complete dark for the sample readings.

An example of the box as seen in the figure below would have multiple layers to it, starting with the bottom where we would place the spectrometer and computer boards with access to fans while the middle layer would be the optical cage setup and the top layer would have the screen for the computer and the cuvette holder opening. The width and length would be about 8 in by 11 in, like a piece of paper, and close to 6 or 7 inches tall. Layer 3 would have a slant to place the screen on for easy ergonomic use of the system as well as the vial hole with a cover. In the prototype, we will make it so when the cover is open the laser will cut off for safety. The only pieces accessible to the user will be the screen and vial cover.

This will also help a lot with the portability of the system to be able to take it out in the field and for technicians to use it with how simple and straightforward the design is. However our only hiccup is the laser we can see that in the images below, the laser is missing from the system this is because the laser that we borrowed from IPS is analog, which means it cannot be controlled by a computer. So, we must have direct access to the front of the laser to control it. Not only that but it is also very large and would make the entire system heavier and less portable. Later on, this is something we could fix by getting a USB-controlled laser that is more compact. For this prototype however we will keep the laser box separate but it will be connected to the system through a fiber connection.

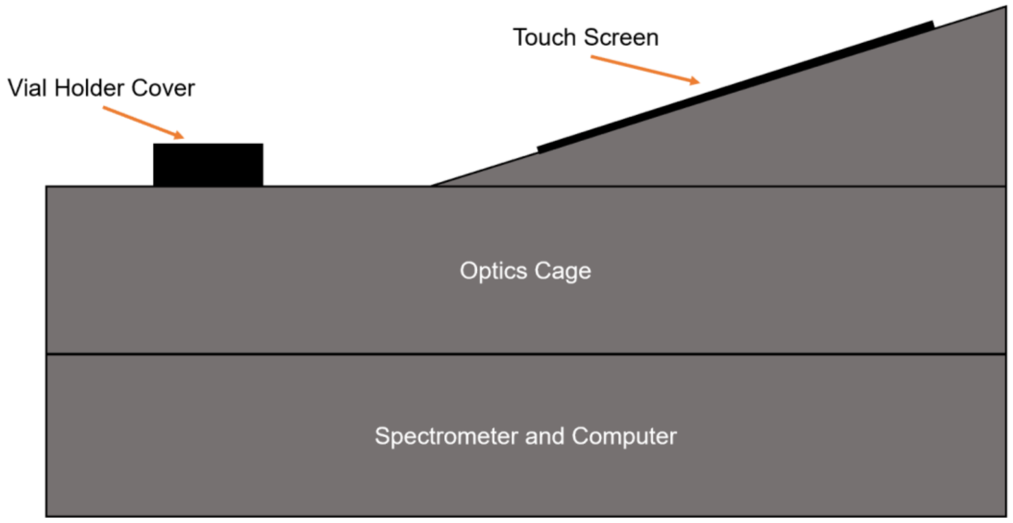


Figure 34. Optical box design example for all 3 layers stacked on top of each other.

To explain each of these levels further, we can see in the image below what will be contained in each section. This box will most likely be made with a 3-D printer as it is the most cost-effective option, and modifications can easily be done to it.

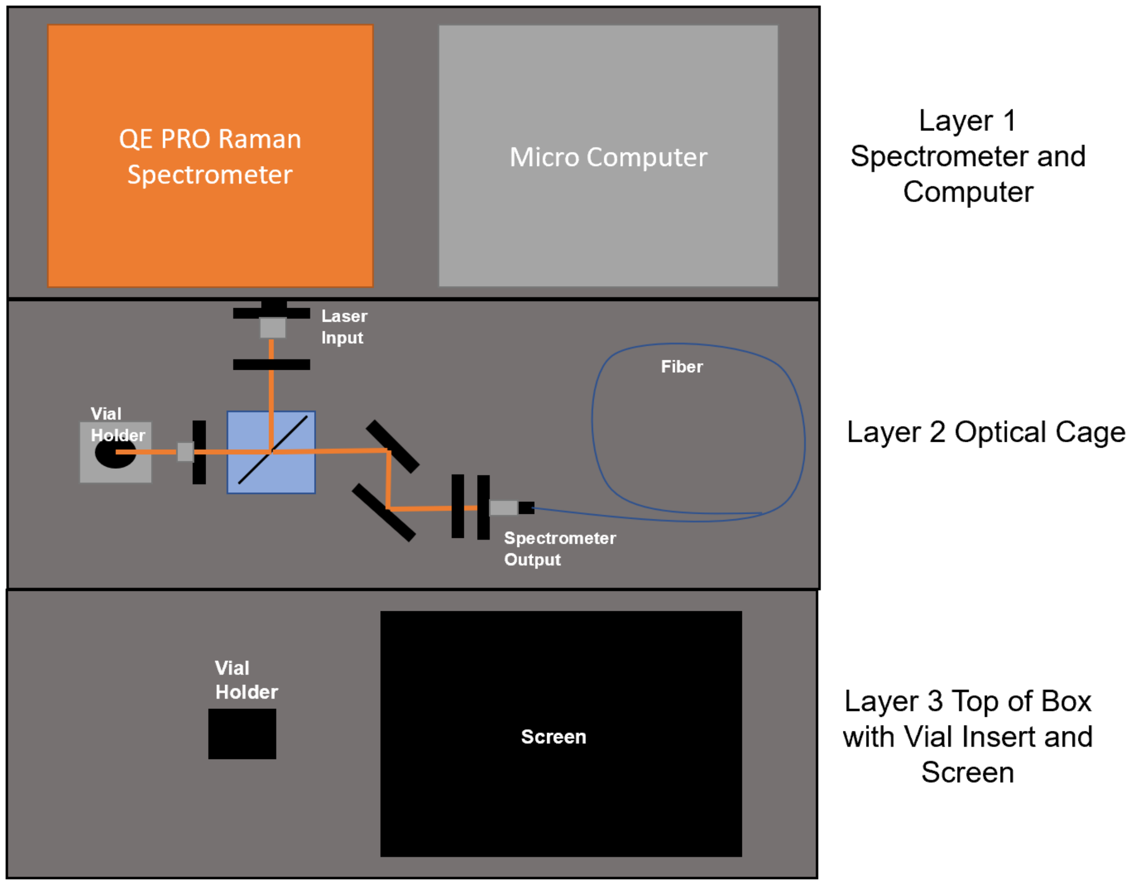

QE PRO Raman Spectrometer
Micro Computer
Layer 1 Spectrometer and Computer
Laser Input
Fiber
Vial Holder
Layer 2 Optical Cage
Spectrometer Output
Vial Holder
Screen
Layer 3 Top of Box with Vial Insert and Screen

Figure 35. The three layers of the optical box are laid out.

5.6.3 Entrance to the cuvette chamber

There are two main designs under consideration for the entrance into the cuvette chamber, a side entrance, or a top entrance with both using different designs for the latch. If we were to make use of a side entrance into the cuvette chamber, we would likely make use of a push to open the hinge. This would allow for a much sleeker and fluid design for the casing of the device. As the push to open hinge makes use of a small magnetic catch or even a series of magnets with the other side having a small section of a steel plate attached to the door. The main issue with using this kind of hinge is the use of a magnet in the device. If the magnet is strong enough the electronic components could become damaged or wipe of the device's programming, which would be disastrous. Another issue is that lasers do not interact with magnets well either, as when a thin magnet is hit by a laser, it can de-magnetize. This could result in the entrance opening during testing and contaminating the test results unless the magnet is properly shielded from the laser. In addition to the latch, additional reinforcement would be added to the edges of the entrance to ensure that no light can interfere with the lasers hitting the sample, while also ensuring that no outside debris or liquids can enter through the entrance as well.

If we were to make the entrance into the cuvette chamber on top of the device, a different design would be used altogether. The overall design would be like a sliding glass door, just with more reinforcement to ensure that no light, debris, or liquids can enter the chamber while not sacrificing the mobility of the entrance. One possible issue with the use of this design would be the wear and tear on the sealant by friction between the latch and material sealing the edges of the opening.

5.6.4 Case Design (materials and shape)

The main material used in the case design will most likely be SECC grade steel, which is steel that has been cold rolled and covered with a coating of zinc via the process of electro galvanization. With the coating of zinc, the steel has a much greater life span when used in devices as the coating prevents the base material from forming oxides of iron, mainly rust. The material's thickness can also range from .4 millimeters to 3 millimeters allowing for a more sturdy and robust case to be constructed. Due to its durability, it is less likely to be scratched, dented, or damaged in comparison to other enclosure materials, such as Aluminum or Acrylonitrile Butadiene Styrene. One of the possible drawbacks is the weight of steel due to its higher density, leading the device to become heavier and less portable. In addition to its weight is the fact that it isn't as good as a heat conductor as aluminum, therefore relying more on how efficient the airflow is in the device.

Aluminum is a good contender for deciding the material to be used primarily since it is so lightweight in comparison to steel due to its lower density, which makes it an ideal choice for a device that is moved often, while also not sacrificing the integrity of the case. It is also is more aesthetically pleasing with its sleek and modern look. It also is resistant to rust corrosion and has built-in flame protection, which could be extremely useful considering what the testing samples will be. Another positive is that aluminum isn't reactive to magnetic forces, meaning that aluminum is an excellent shield against electromagnetic frequency and radio frequency interference. However, there are some downsides to the option as it is much more expensive than steel, and more of the material is needed to get reach the same toughness as a steel casing with less material. As a result, if the amount of aluminum used is little, noise and rattling can be generated by the internal parts. Generally, when aluminum is used in other casings, to keep the cost down, manufacturers will use as little as possible. Therefore, making the cases fragile and prone to dents and scratches, and other damage. Also, although the fact that an aluminum case is naturally radio frequency interference shielded it can be detrimental to the project if Wi-Fi capabilities were desired. Either the Wi-Fi receiver would have to be external or the device would have to be connected via ethernet if the device wanted to be connected to a network.

	SECC Steel	Aluminum	ABS
Arc Resistance (s)	N/A	N/A	60 - 120
Electrical Resistivity (Ω/cm)	.000017	2.69	14×10^{15}
Dielectric Strength (kV/mm)	N/A	N/A	15.7 - 34
Dissipation Factor	N/A	N/A	$50 – 190 \times 10^{-4}$
Linear Temperature Expansion Coefficient (10^{-6}m/(m °C))	10.8 - 12.5	21 - 24	72 - 108
Modulus of elasticity (GPa)	190 - 210	68	1.6 - 2.4
Density (g/cm^3)	7.8	2.7	1.05
Tensile Strength (MPa)	270	90	40 - 50
Price	3 mm thickness By the Ton $720	.1" thickness with dimensions 12" x 12" $29	1/2" thickness with dimensions 48" x 96" $736

Table 10.Comparing the materials that can be used for the case

5.7 Possible Future Hardware Implementations

In this section, the document will discuss possible improvements and variations in the hardware space. These variations are not included with the prototype hardware design, but the hardware is extremely interchangeable for and open-ended, and therefore can be modified and improved upon. Doing so is the industry standard and will help build upon itself to end up being a better overall design. Some of the reasons why we can think of this now but cannot change it is because of either time or funding.

The first interchangeable part would be the touch screen monitor component of the design. As the touch screen is one of the most versatile and important parts, as it is the keystone to user input and displays the results of the device overall. Therefore, its size might need to increase to be able to further display the resulting data that was obtained more clearly. As well as maybe change the type of display between the five main types: Resistive Touch screen, Surface Capacitive Touch screen, Projected Capacitive Touch screen, Surface Acoustic Wave Touch screen, and Infrared Touch screen. The touch screen might also change as different models are introduced as perhaps the parameters and specification might change, such as the intended location of use.

The next changeable part is the single board microcomputer, as the current one uses an intel processor architecture, future iterations of the product might want to use an ARM architecture processor. While both architectures are designed for low-power operation, the ARM architecture is designed to be as simple as possible to keep energy wastage to a minimum and uses Reduced Instruction Set Computing. Whereas Intel architecture is more complex and uses Complex Instruction Set computing. The main difference between the two is that ARM processors use only a single cycle while executing a command, reducing functions. While Intel processors may use a simple command, they must go through several cycles before completing their action. In the future, it might be deemed necessary to add onboard power storage to allow for increased portability and use without having to be connected to a power source. If that is the case, then an ARM processor would be ideal to use as they are designed to use less battery life due to the single-cycle computing. The next improvement that could be made to the design is the capability to save and store graphs or data onto a USB drive. While not as efficient as having a cloud-based database, it is still a marked improvement and allows the transfer of data through a more analog approach. As some companies would prefer to keep a record of the tests that were taken and keep them cataloged allowing them to analyze the data more in-depth and on a grander scale over time.

The final improvement that can be made is making the laser system is not analog and instead controlled by the microcomputer touchscreen interface. This would be mostly to the benefit of the overall design of the system and allow a sleeker and more aesthetically pleasing shell to the product. As the current laser system is analog and must be triggered by the user with a separate switch from the touch screen. This also prevents us from adding safety measures that can be implemented at the software level. Such a measure would ensure that the lasers are not able to engage while the entrance into the cuvette chamber is opened, this would not only save the user power from the wasted firing of the lasers but also help extend the overall life of the prototype by preventing unwanted firings of the lasers. The laser we currently have is called a “fat boy” laser by IPS themselves and because it is quite a big box. On top of this it is analog and not digital we are unable to box it up with the rest of the components making it still portable by not all in one box. Which takes away from the ‘wow’ factor of it being very easy to use. Another issue we run into with the analog is the number on the front of the box is just a number, it does not have a unit attached to it so we don’t know exactly how much mW it is outputting unless we use an optical power meter.

6. Overall Integration and Testing

In this section, we will discuss how different parts of the project come together when testing the components before putting everything together in Senior Design 2. It is crucial to validate that all the components work individually before putting them together. This will also help in case there is an issue later that needs to be fixed, we have a better idea of where the problem could be and how to fix it. Here we start with the optical design components to make sure that most of the optics are working correctly before having a computer system run them.

6.1 Optical System Testing

We were lucky enough as a team to be able to gather a good number of pieces to start building our prototype. Most of this was optical components, as we need to start here before worrying about the software. We decided to go with a cage mounting system for ease of use when building and the rods make sure that everything in the system is always on the same optical plane. Using the cage system with rods as seen in the image below also saves us money and size, as the other option would be to build this system on an optical breadboard.

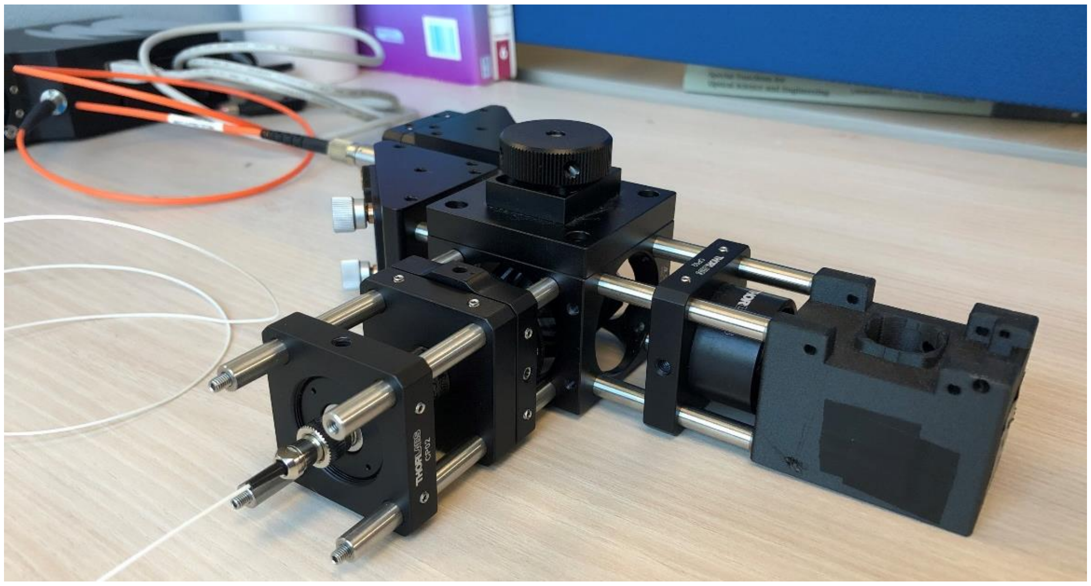


Figure 36. Optical Cage System side view

Once everything was installed on rods, we just needed to align the system. To do this we look at each component in the system and see what it does. The first piece we encounter for alignment is the beam splitter, as the name describes will split the beam acting as a mirror at one angle and a filter at another. We can align the beam splitter by dropping a piece of paper into the vial holder to see where the beam would hit the vial. Once the beam is in the center of the vial holder we can move on to the focusing lens. Here we roughly align the system using the paper method and finding the best focus at the center of the holder, we will come back to this later for minor tweaks. Following the ray path we find the mirrors, if this were a breadboard system we would be able to easily use the paper method here too but because the cage is blocking off access to the actual mirrors we have to plug in the spectrometer with a sample in the vial holder and use the real-time graph to see the peaks. Finding the best alignment here is tricky as we are slowly turning the knobs of the mirror mount while watching the computer screen looking for where the peak looks the best at the highest intensities. We found that the preferred sample to use when aligning is acetone as it gives many very distinct and clear peaks. Once we find the best alignment of the mirrors we can go back to the focusing lens and move that around to see if we can get a better signal, then we realign the mirrors once again as we just changed something in the system. As you would assume working with optics requires a lot of patience and finesse. Below we can see the entirety of the optical system which can be compared to the schematic from earlier.

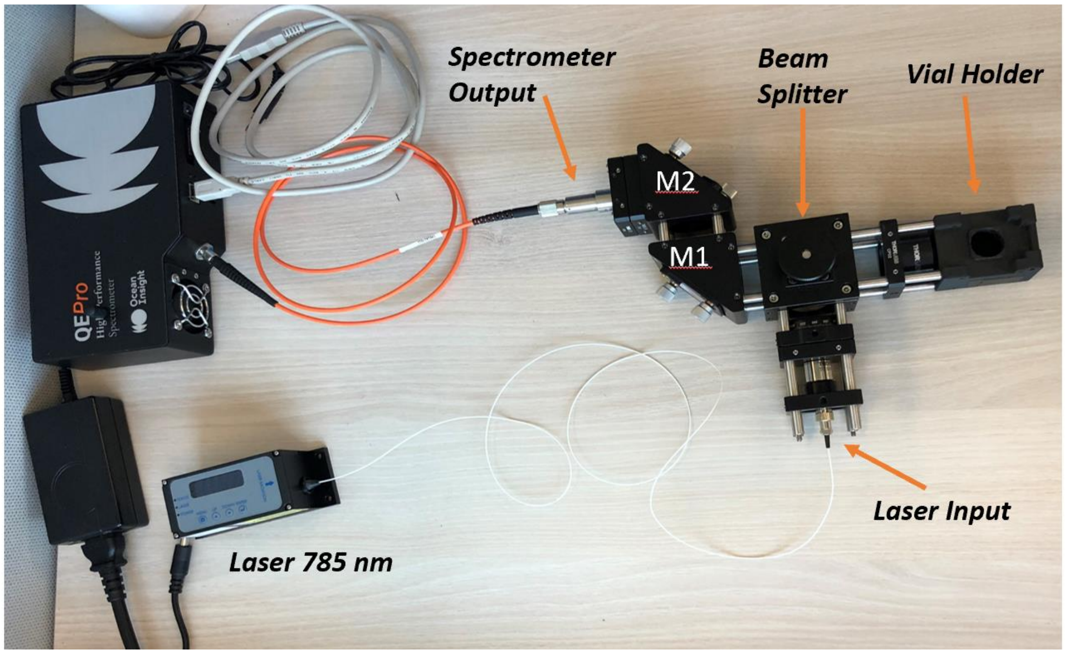


Figure 37. Complete Optical System

All of this preemptive alignment was done with a 785 nm laser that Ocean insight had loaned me as seen above, for the time it was taking IPS to ship the actual laser I would be using, named the “Fat Boy”. When the fat boy laser came in I attached it to my system but the reading on the spectrum was extremely weak. After double-checking my optical cage I tested the laser with an optical power meter and found that the problem was coming from the fiber attached to the laser. This is when I noticed a kink in the fiber where it may have broken. Often optical fibers resemble electrical wires with the way they are coated however it is important to keep in mind that when using an optical fiber it is a glass rod on the inside that is stretched so small it can be curved, it is still easily breakable though if you go past the curve radius set for the fiber. This is what seemed to have happened on this fiber. So, we had to order a new fiber which caused a delay in our testing and integration process as well as costing more money.

Once the system was aligned with the correct fiber, We started to test the Raman power. To start testing the peaks and Raman system I only used the 785 nm laser as I do not have filters for the 680 nm laser yet. For the sample tests I used basic sugar and water, as seen in the figure below, each vial was filled with a different level of sugar and the rest of the vile was filled with purified water. Each of these vials was tested and the data from each was saved and all put onto the same graph. As seen in the figure below, there is a distinct difference in intensity peaks with the different amounts of sugar present in each sample. However, each sample has similar peaks which are good to know that this is the sugar we are testing for each one. This is the start of what we will be continuing in Senior Design 2, we will take data like this and the software will tell you approximately how much sugar is in the sample and if it is sugar. But instead of sugar, we will be detecting water in the fuel. We can easily test fuel for other contaminants like chemicals with the system that we currently have but with the 680 nm laser going into the system this is when we can see the stretch region for water content as explained in previous sections. However, considering we are still only in senior design 1 using sugar is okay.

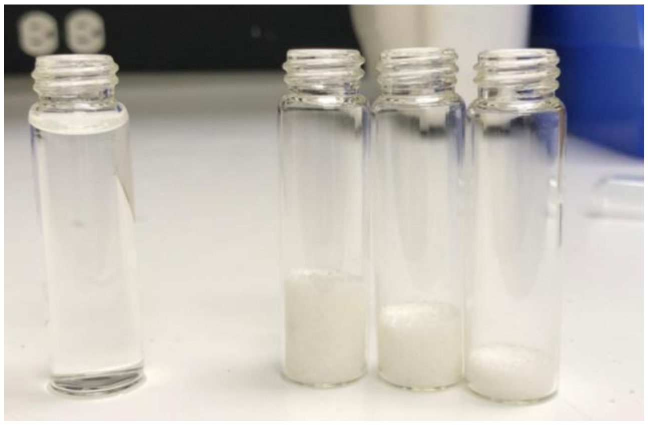


Figure 38. Different levels of sugar in each vial, with water as a control.

As seen in the figure below, there is a distinct difference in intensity peaks with the different amounts of sugar present in each sample. However, each sample has similar peaks which are good to know that this is the sugar we are testing for each one. This is the start of what we will be continuing in Senior Design 2, we will take data like this and the software will tell you approximately how much sugar is in the sample and if it is sugar. But instead of sugar, we will be detecting water in fuel which is where the second laser will come in.

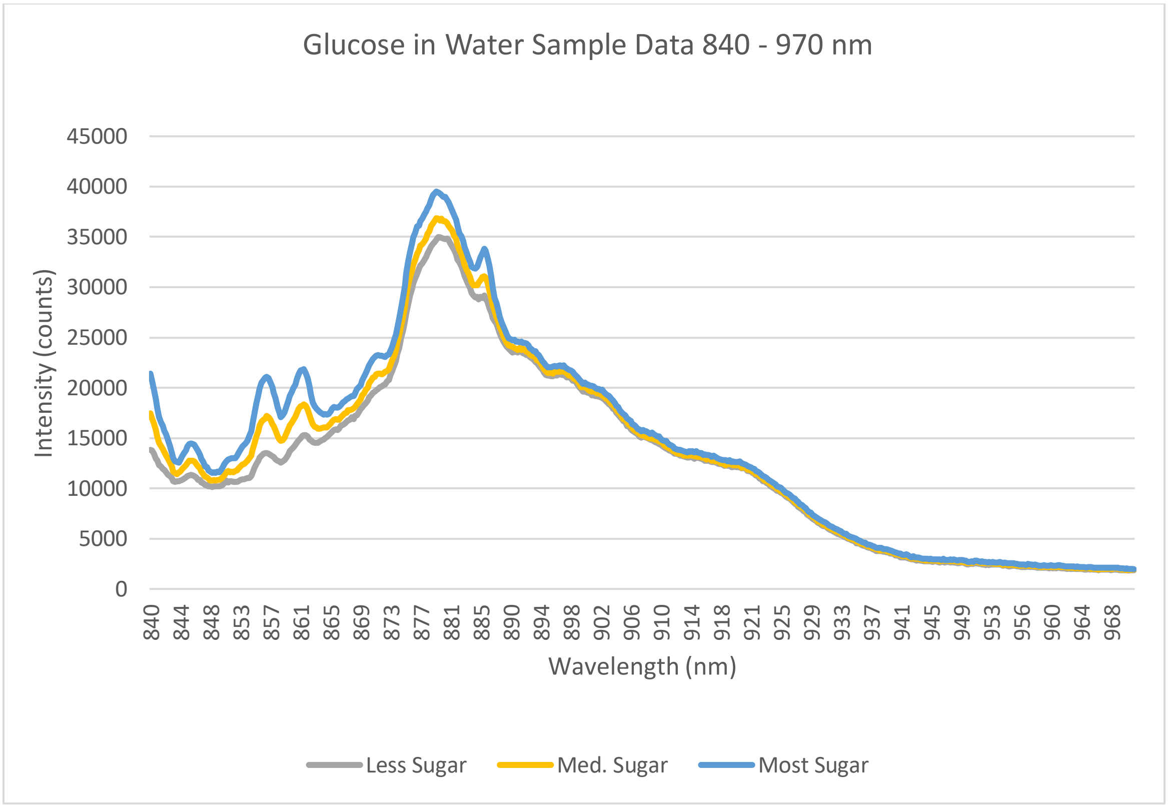


Figure 39. Graph showing different levels of sugar on spectrum graph.

What we have done in this testing is showed that our Raman system works. Normally for a Raman system, this is all you need, usually, you only use a single wavelength laser in the Raman system. We have already achieved something great and we are excited to continue working on this in senior design 2. The only thing that is holding us back is the 680 nm laser filter.

6.2 Software Testing

In this section, the document will discuss how the individual parts and phases of the software flow will be tested and debugged. This will be going through the most commonly used cases to find and fix as many fail states as possible, leading to a more secure and stable final product. This process is a must for any software design and will be done at multiple points throughout the software design process.

6.2.1 Testing the Automated Subroutine

To begin, we will take a look at the automated subroutine and how to go about finding any flaws in the process. The first step of debugging the automated subroutine will be to start the device with full control still given to the designer. The automation if working correctly should start up as soon as it can without any interaction from the user. To add to this, we can use the toast library mentioned in Section 5.2.2 to provide notifications that a Python program is running. This will be needed to ensure this portion of the automation is successful. Unfortunately, we do not have a library like this for the C++ side of the software flow, but with the Python program being below the C++ in the order of execution, if the Python program executes, it is safe to say that the C++ program executed as well. It is important to note that just because these programs execute, does not mean that they are working properly at all. It just ensures that they are executing without human interaction, simplifying the user experience and verifying that the automated subroutine is working correctly. If anything were to fail during this step, the likely culprit will be file naming. The Visual Basic Script uses exact file naming, and if the file is named incorrectly or isn't present in the working directory, the Visual Basic Script will fail to execute it. A likely related culprit could be the registry editor. The registry editor also uses exact file path naming. So these same problems could arise if the Visual Basic Script is not named correctly or is not present in the directory specified in the registry editor. It should be noted once again just to emphasize the importance of it that one needs to use extreme caution in the registry editor. Changing the wrong register while debugging could lead to irreversible damage.

6.2.2 Testing the Main Programs

In this section, we will discuss how to debug the C++ and Python programs. It is important to note when while we are discussing the Python program here, we will not be discussing the Graphical User Interface, as it will have its section, Section 6.2.3. In this section, we will be looking at the C++ program which collects information from the Spectrometer and saves it as a file type, and the detection and recipe algorithms.

For the C++ program, we will first test to make sure that the program is finding the spectrometer correctly. This can be done by checking to see if the spectrometer is being correctly detected by the computer, and can be determined by checking the device manager in Windows 10. Doing this we can find the spectrometer using driver commands and save it as a variable. Next, we will test to see if the data is being collected properly. We can test this by finding potential scenarios that we will already know the readings to. For example, a chemical that is very easily and accurately tested by other means and gives us a strong idea of what the data spread will look like. Then we can test the same chemical with our algorithm and see if we get a similar spread. Most of the potential problems that can occur at this stage will be associated with the spectrometer itself. The main issue we could have is if the program fails to recognize the spectrometer, or if the program can't save the file type for some reason. These problems will need to involve some sort of try-catch block to ensure the entire device does not crash software-wise should one of these problems occur.

For the Python program, we will start with the detection algorithm. Similar to the process of testing the data collection in the C++ program, we analyze scenarios we already know the answer to and see if we get a result similar. We are the ones doing a human detection algorithm when debugging the C++ program and we are just automating this process in the Python program. The program itself should be tested with multiple different chemicals to verify it is working properly. This will also be displayed on the Graphical User Interface but more on that in a later section.

For the recipe algorithm, one can test this by simply presenting the algorithm with data that we already know the recipe too. It is important to note here that we don't need real data for testing the Python side of the software, and just need data in general that it can use. As long as we can find the answer to the data presented ourselves, the data should work for the algorithm. We can present these data sets to the recipe algorithm and test to see if the algorithm gives us a similar result to paper and pencil calculations. These steps should give a strong idea that Artificial Intelligence in a sense is just automating a process that is normally done by humans. We are giving the device a set of rules and it follows those rules the same way we do. Any problems that are found during this step will either be related to the code in a one-off sort of sense. If a problem occurs in a certain spot each time, it is most likely then related to the mathematical equations implemented into the algorithm itself, meaning the algorithm is working correctly, but doesn't use the correct equation, and thus gives the wrong answer.

6.2.3 Testing the Graphical User Interface

The final portion of the software flow to test will be the graphical user interface. Here we are seeing the interaction of all the above moving parts in one place, and thus will be the most intensive to test. This is also the section aside from the collection process where we have the most interaction with non-software-related steps, such as the touchscreen monitor.

The first item to test the graphical user interface will be the option buttons. At first, the graphical user interface should be tested with mouse and keyboard before with touchscreen as this eases the debugging process. The first thing to check will be if the press of the button, meaning do they do anything when interacted with. If yes, then the next item to verify is done they initiate their intended purpose. Does the Collect Dark option collect the baseline data? Does selecting Get Sample to collect a sample from the device? The first test is to make sure the buttons are implemented correctly. The second is to ensure that the functions of the buttons are implemented correctly and work with the graphical user interface.

Continuing from here one can test to see if a waveform appears when the Get Sample option is selected. Here we will see a graph appear each time the sample finishes processing. This will be a start to finish test of the C++ program collecting the data, sending it to the detection algorithm, and then sending the produced graph to the graphical user interface. Any issues at this step will likely be traced back to prior steps, as here we are having the most informative visual of the entire process. Any detectable coding errors should be solvable by looking at the graph portion of the graphical user interface block.

The last thing to test will be the waveform options. These will be the hardest to test of the three and will require prior data collected and processed to compare between. These options will directly affect the data presented so it is important to understand how they will affect it and what will be different. This will also present the interaction of the Python program interacting with C++ in the sense that it is supplying variables.

6.3 Safety Procedures during Testing

During the testing process, caution and safety are paramount when handling such a volatile and flammable liquid as fuel. If the proper precautions are not taken, a mistake could easily result in some of the components being damaged, or even the entire prototype being damaged. In addition to the components being at risk of damage, if safety isn't taken into consideration, then bodily harm could result as well making it even more important to follow our safety precautions.

The first precautionary measure to be taken should be to make sure an excess of vapor isn't produced when we take the desired fuel out of its container and insert the fuel into the vials. Making the use of an exhaust hood or something of a similar function would easily prevent a buildup of vapors in the area, if a hood is not available then ensuring the transfer of fuel to the vial must be done a significant enough of a distance away from the device to ensure safety. Also, making sure excess vapor is not created when translating the fuel to its new container, the vials must be secured properly once the fuel is inserted inside of it. If the vials are not secured properly then vapor could leak from the vial and create a fire hazard in the form of the vapor being near the electrical components in the prototype.

The next precautionary measure is to ensure the use of gloves, goggles, and preferably a mask when handling the transference of fuel from the container into the vials. The ingestion of fuels such as gasoline even in small amounts, as it is primarily comprised of hydrocarbons, can reduce the functionality of the central nervous system and cause organ damage. Not only is it toxic when ingested, but the liquid, as well as the fumes or vapors of the fuel, can cause damage to the skin, eyes, and lungs when inhaled. The combination of gloves, goggles, and a mask helps minimize the possibility of accidental ingestion, inhalation of the toxic vapors and fumes and helps minimize the amount of skin that can come into contact with the fuel.

Since we are using fuels in such a way, it would behoove us to know the symptoms of gasoline poisoning. The symptoms primarily depend on factors such as the length of exposure to the fuel, the age, body weight, and sex of the individual, whether the person touched, swallowed, or inhaled the fuel, the amount of the fuel the person was exposed to, and whether or not they had exposure to other chemicals. Inhaling the vapors from fuels can irritate lung tissues as well as cause several chemicals to be introduced to the bloodstream. Upon entering the bloodstream, these chemicals can make it hard for the body to translate oxygen throughout the body tissues, leading to the tissues dying. The symptoms of gasoline inhalation that are the most common are: blurred vision, staggering, slurred speech, a headache, coughing or wheezing, dizziness or lightheadedness, difficulty breathing, or weakness. While coming into contact with a small amount of fuel on your skin for a short period is considered fairly harmless as skin does not readily absorb a chemical in gasoline, if it remains on the skin or even clothing for several hours it can eventually penetrate the skin. Symptoms of skin and eye exposure to fuel are a pus-like discharge, temporary loss of vision, pain, and discharge if the fuel comes into contact with the eye, first and second-degree burns, skin inflammation, mild skin irritation, and cracking, blistering, or peeling skin. Even though the chance of ingesting gasoline is unlikely we should still be aware of the risks as consuming as few as 2 ounces can cause intoxication, while around 12 ounces can kill a person. Symptoms of ingesting gasoline are slurred speech, drowsiness, heartburn, weakness, blurred vision, vomiting, convulsions, loss of consciousness, confusion, staggering, and heart failure. If someone suspects they have gasoline poisoning, regardless of the variety, should call poison control at 1-800-222-1222, and if the symptoms are severe enough call 911.

The final precautionary measure is to ensure that we have the proper tools to put out a fire if one does occur, preferably in a way that also does not damage our prototype any further. Fuel fires are primarily extinguished through smothering as it deprives the fuel and its vapors of oxygen. A great variety of objects can smother small gasoline fires such as wet rags, woolen cloths, sand, earth, or even ashes. If it is the container of fuel that is on fire, then the container must be cooled with water to lessen the amount of vapor being generated by the fuel. It would be preferable to have a large fire blanket on hand that can cover the entire prototype to deprive the fire of oxygen if one does occur, using an object such as dirt, sand, water, or wet rags to extinguish the fire could potentially further damage the prototype or some of its other major components, such as the laser system, the spectrometer, or the all-in-one motherboard.

The table below is used as a reference for displaying how safe the fuels are to use during the testing process. As the flashpoint is the temperature at which a spark or flame will ignite the vapors created by said fuel. The table shows that the ideal fuels to use for initial testing would be Kerosene, Dodecane, and Diesel fuel. While Gasoline and Hexane are much more volatile and easier to ignite, therefore making our safety precaution measures that much more crucial to follow during the testing process.

Fuel	**Flashpoint (°F)**
Kerosene	100
Dodecane	165
Hexane	-9.4
Diesel	130
Gasoline	-49

Table 11. Flashpoints of different fuel chemicals.

6.3.1 Vial glass options and how they affect safety

The first option for glass vials is made up of Type I glass, which is glass that contains 80% Silica, 10% boric oxide, and small amounts of sodium oxide and aluminum oxide. Due to this composition, it is not chemically reactive and is highly hydrolytic resistant as it contains boric oxide. However, due to its low coefficient of expansion, it is susceptible to a rapid change in temperature. The fact that these vials are chemically reactive, as well as hydrolytic resistant, makes them an ideal glass to comprise our vials, the fact that it has a low coefficient of expansion should not come into play with our testing process so it is a non-issue and can be ignored. The second option for glass vials consists of a Type III glass which consists of untreated soda-lime glass with an average resistance to chemicals. It is comprised of 75% silica, 15% sodium oxide, and small amounts of aluminum oxide, magnesium oxide, and potassium oxide. The aluminum oxide is added to increase chemical durability while the magnesium oxide provides a reduction to the temperature needed during the molding process. As strong chemical durability is needed to withstand the corrosive nature of the fuels. The third option for glass vials is created with Type II glass, these vials are a variant of the Type III glass vial, however, these have a higher hydrolytic resistance because the inner surface of the vials is treated with sulfur. This removes the leachable oxides and prevents blooming and weathering from the bottles. The final option for glass vials is comprised of Type IV glass, which is just general-purpose soda-lime glass. This means that the vials have a low hydrolytic resistance, making the Type IV glass a non-ideal glass to use for our vials as the fuels are rather corrosive.

The main option that stands out is the Type I glass, with the fact that it isn't chemically reactive meaning it won’t taint the samples. The addition that it is hydrolytic resistant also preventing the contamination of our base sample with extended use. Even though it has a low coefficient of expansion, that should not come into play as we do not plan on heating the vials with the fuels in them. Most of the vials that are created with Raman Spectroscopy in mind also consist of Type I glass, making it much easier to source too.

6.4 The Testing Process

To test the prototype, rigorous testing must be conducted to ensure the accuracy of the test results as well as ensuring that the prototype is as robust as possible. The first step in the testing process will be the selection of the testing fuel/fuels. As not only do we need to consider safety but the obtainability of the fuel as well. The table below is used as a reference when used when deciding which of the fuels to use during the testing process, as the chemical formula is one of the reference points to be used for determining the fuel’s ease of obtaining results, as the larger the molecule of the fuel, thus making it easier to observe the response.

Fuel	Chemical formula
Kerosene	C_6H_y - $C_{16}H_y$
Dodecane	$C_{12}H_{26}$
Hexane	C_6H_{14}
Diesel	$C_{12}H_x$ - $C_{20}H_y$
Gasoline	C_xH_y

Table 12. Chemical Formulas for different chemicals of Fuel

After taking into consideration both the flashpoint and the chemical formula of the various fuels that we have to choose from, we believe that dodecane would be the ideal introductory fuel as a test sample. Not only does its large molecule size make it one of the more ideal fuels to observe the response of the Raman laser, but it has the highest flashpoints of all the fuel options that we have to choose from making it also the safest fuel. After we rigorous testing and introductory calibrating of our prototype, we can expand to using the other fuels to ensure that our prototype is as robust and diverse as possible, with ideally using diesel fuel as the second testing fuel as it is one of the more varied fuels as it contains paraffin, aromatics, and naphthenes, while also still ensuring fire safety with its high flashpoint.

For the start of the testing process, we should make sure that the initial tests start with the purest form of the samples that we can obtain, as the initial tests will be used heavily as a reference point for testing. It would behoove us to also have previous results of other tests of using Raman Spectroscopy on fuels to be able to compare our results to ensure accuracy. Ensuring that our initial testing samples are pure should result in easy-to-read results and acts as a good starting point for the calibration of our prototype as there will be no contaminants to alter our results.

After ensuring that the prototype can obtain accurate results from pure fuel samples, the next step would be to create various mixtures by taking the pure forms of the fuel and combining them with varying concentrations of water to ensure that our prototype can accurately distinguish between the fuel and water and be able to ascertain the concentrations of both. While this isn't a very complicated test, it is still a good one to conduct early as calibrating the system to be able to distinguish between fuel and water and be able to ascertain the concentrations of both substances is important. We will start testing concentrations of the fuel and water in variations of tens. Following the testing phases below. If the system can state the concentration of both substances to an accuracy of ±2%, then we can consider it a pass.

Test	Water concentration	Fuel Concentration
1	0 %	100%
2	10%	90%
3	20%	80%
4	30%	70%
5	40%	60%
6	50%	50%
7	60%	40%
8	70%	30%
9	80%	20%
10	90%	10%
11	100%	0%

Table 13. Water concentration vs Fuel concentration

If the system can accurately predict mixtures of fuel and water in these greater concentrations, we can then change the ratios to smaller concentrations to increments in 5% and even further down the line to 1% to test our prototype efficiently and observe the results. Of course, if the increments are smaller the accept variances in the accuracy will lower as well, this will also help in the calibration of our prototype.

The final testing phase will involve contaminating the water and fuel mixtures with various bacterial and molds to ensure that we can detect the contaminants, aka the bacteria and mold, while also still giving accurate results of the concentration of water in the fuel. Being able to detect that there is bacterial contained in the sample, regardless of the concentration of it would be the first step as identifying the existence of the bacteria in the fuel and water mixture is key to our prototype's goal. Ideally, we will start with a greater concentration of bacteria and then slowly decrease the amount in samples to ensure that we can detect the contaminants and the amount that exists in the sample. The final tests will consist of fuel that was initially pure but then stored in a container for an unspecified amount of time. Allowing it to be contaminated with an unknown amount of water and preferably several types of bacteria and mold. If we can accurately analyze the sample and display the correct ratio of water to fuel as well as determine that there is bacteria or mold in the water, we can consider the prototype testing phase success and move onto the “stretch goals” of our prototype.

7. Administration

In this section, we will be talking about all the administrative content through time management with the milestone discussion and how the work was split up amongst the team. This will also include our budgeting for this project.

7.1 Milestones

In the table below, we see the milestones and the status of their completion. It starts from our first week in Senior Design 1 and calculates the timing for the future weeks ahead into Senior Design 2.

Week	Date	Description	Status
Senior Design 1			
1	1/11/2021	Project Idea / Meeting group members	Done
2	1/18/2021	Talking with Sponsor / Project Idea	Done
3	1/25/2021	Initial project documentation	Done
4	2/1/2021	Start ordering parts	Done
5	2/8/2021	Updated D&C	Done
6	2/15/2021	Initial Design Stage	Done
7	2/22/2021	Testing spectrometer and software	Done
8	3/1/2021		Done
9	3/8/2021	Hardware Testing	Done
10	3/15/2021		Done
11	3/22/2021	Prototype Planning	Done
12	3/29/2021	60-page draft	Done
13	4/5/2021	Prototype Planning	Done
14	4/12/2021	Final Draft	Done
15	4/19/2021	Final Document	Done
Senior Design 2			
16	5/17/2021	Building Prototype	In Process
17	5/24/2021		In Process
18	5/31/2021		In Process
19	6/7/2021	Testing and Redesign	In Process
20	6/14/2021		In Process
21	6/21/2021		In Process
22	6/28/2021	Finalizing Prototype	In Process
23	7/5/2021		In Process
24	7/12/2021		In Process
25	7/19/2021	Peer Presentation	In Process
26	7/26/2021	Final Report	In Process
27	8/2/2021	Final Presentation	In Process

Table 14. Milestones and Dates

7.2 Budget

As a team, we were fortunate enough to have Ocean Insight sponsor most of this project with many of the optical parts and most important, the spectrometer. As well as Innovative Photonic Solutions (IPS) for letting us borrow the dual-wavelength laser system. This project would've been much more difficult without the help of IPS and their laser as it combines the exact two wavelengths, we need for the laser input. We have provided the budget sheet in the table below with everything used for the most part and the cost accordingly. This was approved by our main sponsor Ocean Insight. However, the table below is not the last version of the budget as things are changed and added to the design these numbers will change. This provides an overview budget of the project.

Description	Quantity	Cost / Unit	Total Cost
Raman Spectrometer	1	$15,000.00	$15,000.00
Laser (680 & 785 nm)	1	$13,500.00	$13,500.00
Optical Cage Components	1	$1,000.00	$1,000.00
Optical Fiber	3	$500.00	$1,500.00
Fiber Coupler	1	$100.00	$100.00
Collimating Lens	1	$150.00	$150.00
Fiber Holder	2	$100.00	$200.00
Focusing Lens	2	$150.00	$300.00
Filter	2	$500.00	$1,000.00
Computer / Software	1	$800.00	$800.00
Power Converter	1	$50.00	$50.00
Samples	1	$100	$100.00
Cuvettes	1	$100.00	$100.00
Box	1	$400.00	$400.00
Total Cost			**$35,700**

Table 15. Budget overview

The most crucial part of this budget is that it contains many parts to our design that are expensive to replace if something were to happen to them. In the case that something was to happen to one of our parts it is not only a problem for this senior design team but our sponsors as well.

8. Conclusion

RSCL began as just an idea, something that is not on the market yet, something that can solve many problems in fuel. Through this project we made these ideas come to life backed by knowledge and research. We designed a noninvasive way to measure water contamination in fuel through Raman Spectroscopy, not only do we now have an extremely efficient and reliable way to detect water content but there is no need to alter the sample in any way to do so. As we learned through this project, it is easy to detect substances like chemicals in water, but it is very difficult to find water content in substances. This RSCL is the answer to that. Not only can it be used for fuel, but it can be used later for different chemicals that are used in batteries or substances that are reactive to water.

We created a Raman spectroscopy concatenated laser system. With the help of Innovative Photonic Solutions (IPS), we were given a laser system that did both the 785 nm and 680 nm wavelengths needed for this project. Without their help, we would've had to build a secondary optical system just to be able to combine the lasers. Their generosity has been invaluable to this project. IPS owns the patent for a dual-wavelength Raman Probe, so to have their node of approval is very appreciated. The guidance and sponsorship by Ocean Insight are just as invaluable, they continued to push our project to the boundaries while making it financially possible to create the RSCL. Ocean Insight provided us with our most expensive piece of equipment, the QE PRO Raman spectrometer as well as many of the components in this project. All the pieces in the RSCL were decided on safety, cost efficiency, and being environmentally friendly. The end goal for this project is to make so this system will be portable, so technicians can easily use it directly to test fuel or other chemicals on-site in a matter of minutes. No need to have long wait times or send the samples to a lab.

The purpose of this project is to create an efficient way to find water in fuel and later more chemicals. Although our world is moving towards battery-powered systems in a way to eliminate fuel and pollution, we still use fuel a lot and it will not be replaced that quickly especially for large heavy machinery and equipment. The RSCL will help with the environmentally friendly movement by not letting fuel go to waste and in turn saving engines. If we can check the storage tanks more often it is likely that we can save the fuel before it goes bad or find a way to reverse the damage. As well as always being prepared for natural disaster relief.

We wanted to do this project because it not only challenged everyone in the team to do stretch our field into something, we are not normally used to but to also make a system that is not available to markets yet. This project is not only improving Raman Spectroscopy, it will guide a new path for water detection systems.

Appendices

In this section will see the references used for this paper, in the sections according to software, optical, or hardware. We will also see the copyright permissions to use the images we have used in this paper.

Appendix A – References

Software References

[1] "AML-S905X-CC (Le Potato)," *Libre Computer*, 26-Jun-2018. [Online]. Available: https://libre.computer/products/boards/aml-s905x-cc/. [Accessed: 23-Apr-2021].

[2] "Autorun a Python script on windows startup," *GeeksforGeeks*, 26-Feb-2019. [Online]. Available: https://www.geeksforgeeks.org/autorun-a-python-script-on-windows-startup/. [Accessed: 23-Apr-2021].

[3] "The C++ Resources Network," *cplusplus.com*. [Online]. Available: https://www.cplusplus.com/. [Accessed: 23-Apr-2021].

[4] "Coding Standards For Quality and Compliance," *Perforce*. [Online]. Available: https://www.perforce.com/resources/qac/coding-standards. [Accessed: 23-Apr-2021].

[5] "Community," *Chocolatey Software*. [Online]. Available: https://community.chocolatey.org/. [Accessed: 23-Apr-2021].

[6] Corob-Msft, "Build and run a C++ console app project," *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/cpp/build/vscpp-step-2-build?view=msvc-160. [Accessed: 23-Apr-2021].

[7] Dev Ed, "Build A Python GUI App Tutorial," *YouTube*, 19-Oct-2019. [Online]. Available: https://www.youtube.com/watch?v=jE-SpRI3K5g. [Accessed: 23-Apr-2021].

[8] "Enterprise Open Source and Linux," *Ubuntu*. [Online]. Available: https://ubuntu.com/. [Accessed: 23-Apr-2021].

[9] *GIMP*. [Online]. Available: https://www.gimp.org/. [Accessed: 23-Apr-2021].

[10] J. R. Jacob, "How to make Windows 10 Toast Notifications with Python," *Medium*, 26-Jan-2018. [Online]. Available: https://towardsdatascience.com/how-to-make-windows-10-toast-notifications-with-python-fb3c27ae45b9. [Accessed: 23-Apr-2021].

[11] *java.com*. [Online]. Available: https://www.java.com/en/download/manual.jsp. [Accessed: 23-Apr-2021].

[12] K. Foundation, "Digital Painting. Creative Freedom.," *Krita*, 31-Aug-2020. [Online]. Available: https://krita.org/en/. [Accessed: 23-Apr-2021].

[13] "LattePanda Delta 432 – Tiny Ultimate Windows / Linux Device 4GB/32GB," *DFRobot*. [Online]. Available: https://www.dfrobot.com/product-1908.html. [Accessed: 23-Apr-2021].

[14] “LattePanda Delta 432,” *LattePanda*. [Online]. Available: https://www.lattepanda.com/products/lattepanda-delta-432.html. [Accessed: 23-Apr-2021].

[15] “Libre Computer Board AML-S905X-CC (Le Potato) 2GB 64-bit Mini Computer for 4K Media,” *Amazon*. [Online]. Available: https://www.amazon.com/Libre-Computer-AML-S905X-CC-Potato-64-bit/dp/B074P6BNGZ/ref=sr_1_1?dchild=1&keywords=%22libre+computer%22+aml-s905x-cc&linkCode=sl2&linkId=6ef3196e613636af4f5bb7dfcd2d9935&qid=1619204469&sr=8-1. [Accessed: 23-Apr-2021].

[16] M. Huculak, “Running your first batch script on Windows 10,” *Windows Central*, 16-Oct-2020. [Online]. Available: https://www.windowscentral.com/how-create-and-run-batch-file-windows-10. [Accessed: 23-Apr-2021].

[17] “Microsoft Windows 10 Home | Download,” *Amazon*. [Online]. Available: https://www.amazon.com/Microsoft-Windows-10-Home-Download/dp/B01019BM7O/ref=sr_1_3?dchild=1&keywords=windows+10&qid=1619204726&sr=8-3. [Accessed: 23-Apr-2021].

[18] “Microsoft,” *Microsoft Support*. [Online]. Available: https://support.microsoft.com/en-us/windows/how-to-open-registry-editor-in-windows-10-deab38e6-91d6-e0aa-4b7c-8878d9e07b11. [Accessed: 23-Apr-2021].

[19] *MyPaint*. [Online]. Available: http://mypaint.org/. [Accessed: 23-Apr-2021].

[20] “OmniDriver and SPAM,” *OmniDriver & Spam | Ocean Insight*. [Online]. Available: https://www.oceaninsight.com/products/software/drivers/omnidriver-and-spam/. [Accessed: 23-Apr-2021].

[21] “os - Miscellaneous operating system interfaces,” *os - Miscellaneous operating system interfaces - Python 3.9.4 documentation*. [Online]. Available: https://docs.python.org/3/library/os.html. [Accessed: 23-Apr-2021].

[22] *Project Jupyter*. [Online]. Available: https://jupyter.org/. [Accessed: 23-Apr-2021].

[23] “PyCharm: the Python IDE for Professional Developers by JetBrains,” *JetBrains*. [Online]. Available: https://www.jetbrains.com/pycharm/. [Accessed: 23-Apr-2021].

[24] “Python 3.0 Release,” *Python.org*. [Online]. Available: https://www.python.org/download/releases/3.0/. [Accessed: 23-Apr-2021].

[25] Raspberry Pi, “Buy a Raspberry Pi Zero W,” *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-zero-w/. [Accessed: 23-Apr-2021].

[26] S. Brink, “Add VBScript File to New Context Menu in Windows 10,” *Windows 10 Help Forums RSS*, 30-May-2020. [Online]. Available: https://www.tenforums.com/tutorials/8011-add-vbscript-file-new-context-

menu-windows-10-a.html#:~:text=A%20.,vb%20file%20extension. [Accessed: 23-Apr-2021].
[27] S. Team, *Home - Spyder IDE*. [Online]. Available: https://www.spyder-ide.org/. [Accessed: 23-Apr-2021].
[28] “seabreeze,” *python*. [Online]. Available: https://python-seabreeze.readthedocs.io/en/latest/. [Accessed: 23-Apr-2021].
[29] T. Scott, “FizzBuzz: One Simple Interview Question,” *YouTube*, 31-Jul-2017. [Online]. Available: https://www.youtube.com/watch?v=QPZ0pIK_wsc. [Accessed: 23-Apr-2021].
[30] “tkinter - Python interface to Tcl/Tk,” *tkinter - Python interface to Tcl/Tk - Python 3.9.4 documentation*. [Online]. Available: https://docs.python.org/3/library/tkinter.html. [Accessed: 23-Apr-2021].
[31] “Visual Studio IDE, Code Editor, Azure DevOps, & App Center,” *Visual Studio*, 22-Apr-2021. [Online]. Available: https://visualstudio.microsoft.com/. [Accessed: 23-Apr-2021].
[32] W. Gay, “Advanced Raspberry Pi: Raspbian Linux and GPIO Integration,” *Amazon*, 2018. [Online]. Available: https://www.amazon.com/Raspberry-Pi-Zero-Wireless-model/dp/B06XFZC3BX. [Accessed: 23-Apr-2021].
[33] “win10toast,” *PyPI*. [Online]. Available: https://pypi.org/project/win10toast/. [Accessed: 23-Apr-2021].
[34] “The World's Most Popular Data Science Platform,” *Anaconda*. [Online]. Available: https://www.anaconda.com/. [Accessed: 23-Apr-2021].

Optical References

[35] “ANSI Z136.1 - Safe Use of Lasers,” *The Laser Institute*, 08-Oct-2020. [Online]. Available: https://www.lia.org/resources/laser-safety-information/laser-safety-standards/ansi-z136-standards/z136-1. [Accessed: 27-Apr-2021].
[36] B. Performance, “Fuel additives: Diesel additives: Oil additives: Bell performance.” [Online]. Available: https://www.bellperformance.com/. [Accessed: 26-Apr-2021].
[37] By: Chad Christiansen Product Quality and Additives Manager in Agriculture and Farming and C. Christiansen, “Warning signs your diesel is water-contaminated.” [Online]. Available: https://www.cenex.com/about/cenex-information/cenexperts-blog-page/agriculture-and-farming/water-contaminated-diesel. [Accessed: 26-Apr-2021].
[38] Center for Devices and Radiological Health, “Important Information for Laser Pointer Manufacturers,” *U.S. Food and Drug Administration*. [Online]. Available: https://www.fda.gov/radiation-emitting-products/laser-products-and-instruments/important-information-laser-pointer-manufacturers. [Accessed: 27-Apr-2021].

[39] Center for Devices and Radiological Health, “Laser Products Guidance - IEC 60825-1 Ed. 3 and IEC 60601-2-22 Ed. 3.1,” *U.S. Food and Drug Administration*. [Online]. Available: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/laser-products-conformance-iec-60825-1-ed-3-and-iec-60601-2-22-ed-31-laser-notice-no-56. [Accessed: 27-Apr-2021].

[40] “Characterization of gasoline BY Raman spectroscopy with Chemometric analysis.” [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00032719.2016.1210616. [Accessed: 27-Apr-2021].

[41] H. Ge, Z. Ye, and R. He, “Raman spectroscopy of diesel and gasoline engine-out Soot using different laser power,” *Journal of Environmental Sciences*, vol. 79, pp. 74–80, 2019.

[42] J. Kiefer, “Dual-Wavelength Raman Fusion Spectroscopy,” *Analytical Chemistry*, vol. 91, no. 3, pp. 1764–1767, 2019.

[43] J. Kiefer, “Dual-Wavelength Raman Spectroscopy: Improved Compactness and Spectral Resolution,” *American Pharmaceutical Review*, Oct. 2014.

[44] J. Tedesco, “785-nm laser benefits Raman spectroscopy,” 01-Sep-2000. [Online]. Available: https://www.laserfocusworld.com/test-measurement/test-measurement/article/16555494/785nm-laser-benefits-raman-spectroscopy. [Accessed: 27-Apr-2021].

[45] J. Wei, A. Wang, and K. Connor, “COMPARING RAMAN SIGNAL STRENGTHS OF BIOMARKERS AND MINERALS MEASURED BY A MULTI-WAVELENGTH RAMAN SYSTEM ,” *Lunar and Planetary Science Conference*, no. 46.

[46] L. T. Kerr, H. J. Byrne, and B. M. Hennelly, “Optimal choice of sample substrate and laser wavelength for Raman spectroscopic analysis of biological specimen,” *Analytical Methods*, vol. 7, no. 12, pp. 5041–5052, 2015.

[47] M. P. Arroyo, T. P. Birbeck, D. S. Baer, and R. K. Hanson, “Dual diode-laser fiber-optic diagnostic for water-vapor measurements,” *Optics Letters*, vol. 19, no. 14, p. 1091, 1994.

[48] M. T. Meyer, “How Does Concatenation Enhance Raman Spectroscopy?,” *RPMC Lasers Blog*. [Online]. Available: https://blog.rpmclasers.com/how-does-concatenation-enhance-raman-spectroscopy. [Accessed: 27-Apr-2021].

[49] P. Edmonds and J. J. Cooney, “Identification of Microorganisms isolated from jet fuel systems,” *Applied Microbiology*, vol. 15, no. 2, pp. 411–416, 1967.

[50] “Raman: Wavelength Matters,” *Wasatch Photonics*, 16-Jun-2020. [Online]. Available: https://wasatchphotonics.com/technologies/raman-spectroscopy-wavelength-matters/. [Accessed: 27-Apr-2021].

[51] *RamanBasics*. [Online]. Available: https://www.sas.upenn.edu/~crulli/RamanBasics.html. [Accessed: 27-Apr-2021].

[52] Source, “New, less-expensive testing Method approved for Containment Sumps,” 01-Jun-2017. [Online]. Available: https://www.sourcena.com/sourceline/new-less-expensive-testing-method-approved-containment-sumps/. [Accessed: 26-Apr-2021].

[53] Sponsored by B&W TekJan 25 2021, “Correct laser wavelength for raman material identification,” 25-Jan-2021. [Online]. Available: https://www.azom.com/article.aspx?ArticleID=11871. [Accessed: 27-Apr-2021].

[54] T. Hill, “Microbial growth in aviation fuel,” *Aircraft Engineering and Aerospace Technology*, vol. 75, no. 5, pp. 497–502, 2003.

[55] T. Hill, “Microbial growth in aviation fuel,” *Aircraft Engineering and Aerospace Technology*, vol. 75, no. 5, pp. 497–502, 2003.

[56] Y. Mattley, “Characterization of Diesel Fuel Using a Modular Raman System,” *Spectroscopy -Springfield then Eugene then Duluth-* , vol. 29, Dec. 2014.

[57] Z136.2 - Safe Use of Optical Fiber Communication Systems Utilizing Laser Diode and LED Sources,” *The Laser Institute*, 29-Aug-2017. [Online]. Available: https://www.lia.org/resources/laser-safety-information/laser-safety-standards/ansi-z136-standards/z136-2. [Accessed: 27-Apr-2021].

Hardware References

[58] D. Strain, “Lasers make magnets behave like fluids,” *ScienceDaily*, 18-Apr-2019. [Online]. Available: https://www.sciencedaily.com/releases/2019/04/190418094257.htm. [Accessed: 26-Apr-2021].

[59] The Editors of Encyclopaedia Britannica, “Flash point,” *Encyclopædia Britannica*, 15-Oct-2012. [Online]. Available: https://www.britannica.com/science/flash-point. [Accessed: 26-Apr-2021].

[60] The Editors of Encyclopaedia Britannica, “Kerosene,” *Encyclopædia Britannica*, 19-Apr-2012. [Online]. Available: https://www.britannica.com/science/kerosene. [Accessed: 26-Apr-2021].

[61] J. Hanania, J. Donev, K. Stenhouse, J. Jenden, and B. Heffernan, “Kerosene,” *Kerosene - Energy Education*, 29-Aug-2017. [Online]. Available: https://energyeducation.ca/encyclopedia/Kerosene. [Accessed: 26-Apr-2021].

[62] J. Huizen, “Gasoline and health effects: Symptoms and treatment,” *Medical News Today*, 23-Oct-2018. [Online]. Available: https://www.medicalnewstoday.com/articles/323426#causes. [Accessed: 26-Apr-2021].

[63] no author listed, “Acrylonitrile Butadiene Styrene (ABS) and its Features,” *Acrylonitrile Butadiene Styrene (ABS Plastic): Uses, Properties & Structure*. [Online]. Available: https://omnexus.specialchem.com/selection-guide/acrylonitrile-butadiene-styrene-abs-plastic. [Accessed: 26-Apr-2021].

[64] no author listed, “Aluminium: Specifications, Properties, Classifications and Classes,” *AZoM.com*, 17-May-2005. [Online]. Available: https://www.azom.com/article.aspx?ArticleID=2863. [Accessed: 26-Apr-2021].

[65] no author listed, “Diesel and Gasoline,” *AMF*. [Online]. Available: https://www.iea-amf.org/content/fuel_information/diesel_gasoline. [Accessed: 26-Apr-2021].

[66] No author listed, “Diesel Fuel Test Kits & Water in Fuel Test Kits,” *Dieselcraft*, 23-Apr-2021. [Online]. Available: https://dieselcraft.com/fuel-test-kits/. [Accessed: 26-Apr-2021].

[67] No author listed, “Dodecane,” *National Center for Biotechnology Information. PubChem Compound Database*, 16-Sep-2004. [Online]. Available: https://pubchem.ncbi.nlm.nih.gov/compound/Dodecane#section=Flash-Point. [Accessed: 26-Apr-2021].

[68] no author listed, “Five Types of Touch Screen Monitor Technology; Which Is Best For You?,” *TRU*, 18-Nov-2020. [Online]. Available: https://tru-vumonitors.com/touch-screen-basics/. [Accessed: 26-Apr-2021].

[69] No author listed, “Guide to eDP (Embedded Display Port) Cables,” *Quadrangle Products Inc.*, 11-Feb-2021. [Online]. Available: https://www.quadrangleproducts.com/guide-to-edp-embedded-display-port-cables/. [Accessed: 26-Apr-2021].

[70] no author listed, “Hexane,” *National Center for Biotechnology Information. PubChem Compound Database*, 16-Sep-2004. [Online]. Available: https://pubchem.ncbi.nlm.nih.gov/compound/Hexane. [Accessed: 26-Apr-2021].

[71] No author listed, “TO PUT OUT GASOLINE FIRES.,” *Fire Engineering*, 03-Sep-2019. [Online]. Available: https://www.fireengineering.com/leadership/to-put-out-gasoline-fires/. [Accessed: 26-Apr-2021].

[72] no author listed, “Types of A/D Converters - The Ultimate Guide,” *Dewesoft*, 22-Apr-2020. [Online]. Available: https://dewesoft.com/daq/types-of-adc-converters#adc-technologies. [Accessed: 26-Apr-2021].

[73] Pharmapproach, “Glass Containers for Pharmaceutical Use: Composition, Types, Evaluation,” *Pharmapproach.com*, 29-Feb-2020. [Online]. Available: https://www.pharmapproach.com/glass-containers-for-pharmaceutical-use/. [Accessed: 26-Apr-2021].

[74] S. M. W. Imam, “SECC Steels - Importance in Manufacturing Industries,” *LinkedIn*, 19-Nov-2016. [Online]. Available: https://www.linkedin.com/pulse/secc-steels-importance-manufacturing-industries-sm-waqas-imam. [Accessed: 26-Apr-2021].

[75] S. Stöckel, J. Kirchhoff, U. Neugebauer, P. Rösch, and J. Popp, “The application of Raman spectroscopy for the detection and identification of microorganisms,” *Analytical Science Journals*, 07-Dec-2015. [Online]. Available:

https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/full/10.1002/jrs.4844. [Accessed: 26-Apr-2021].

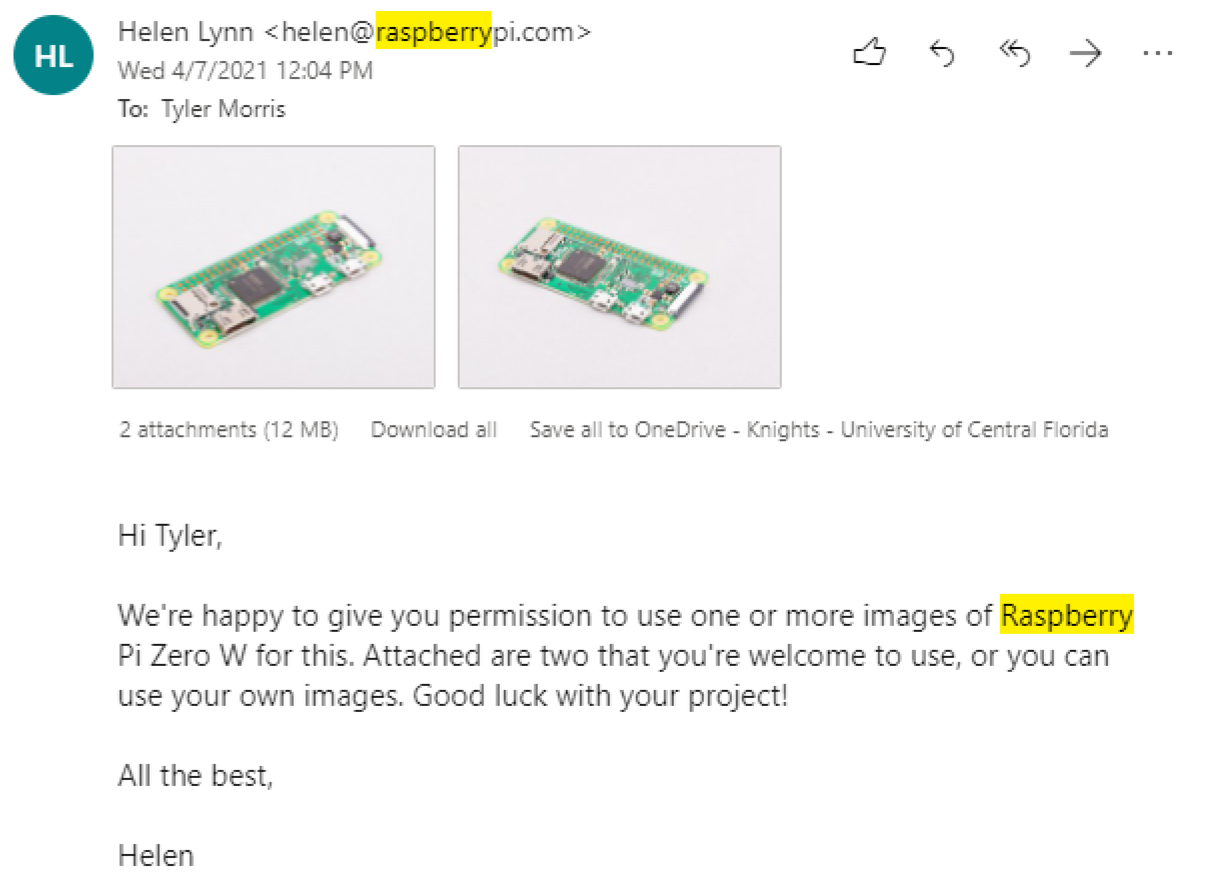
[76] no author listed, “What Is The Flashpoint Of Gasoline Vs. Diesel Fuel?,” *Kendrick Oil*, 24-Feb-2020. [Online]. Available: https://kendrickoil.com/what-is-the-flashpoint-of-gasoline-vs-diesel-fuel/. [Accessed: 27-Apr-2021].

[77] no author listed, *UL 94*. Northbrook, Illinois: Underwriters Laboratories, 2012.

[78] no author listed, *UL 796, 8th edition*. Northbrook, Illinois: Underwriters Laboratories, 2012.

[79] P. Larkin, *IR and Raman Spectroscopy: Principles and Spectral Interpretation*. New York, New York: Elsevier, 2011.

Appendix B – Copyright Permissions



HL

Helen Lynn <helen@raspberrypi.com>
Wed 4/7/2021 12:04 PM
To: Tyler Morris

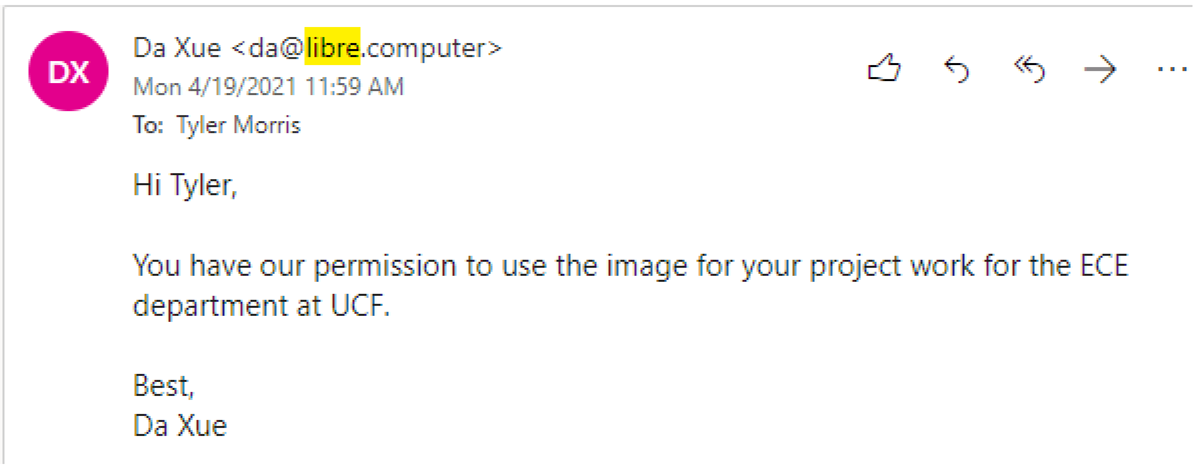
2 attachments (12 MB) Download all Save all to OneDrive - Knights - University of Central Florida

Hi Tyler,

We're happy to give you permission to use one or more images of Raspberry Pi Zero W for this. Attached are two that you're welcome to use, or you can use your own images. Good luck with your project!

All the best,

Helen



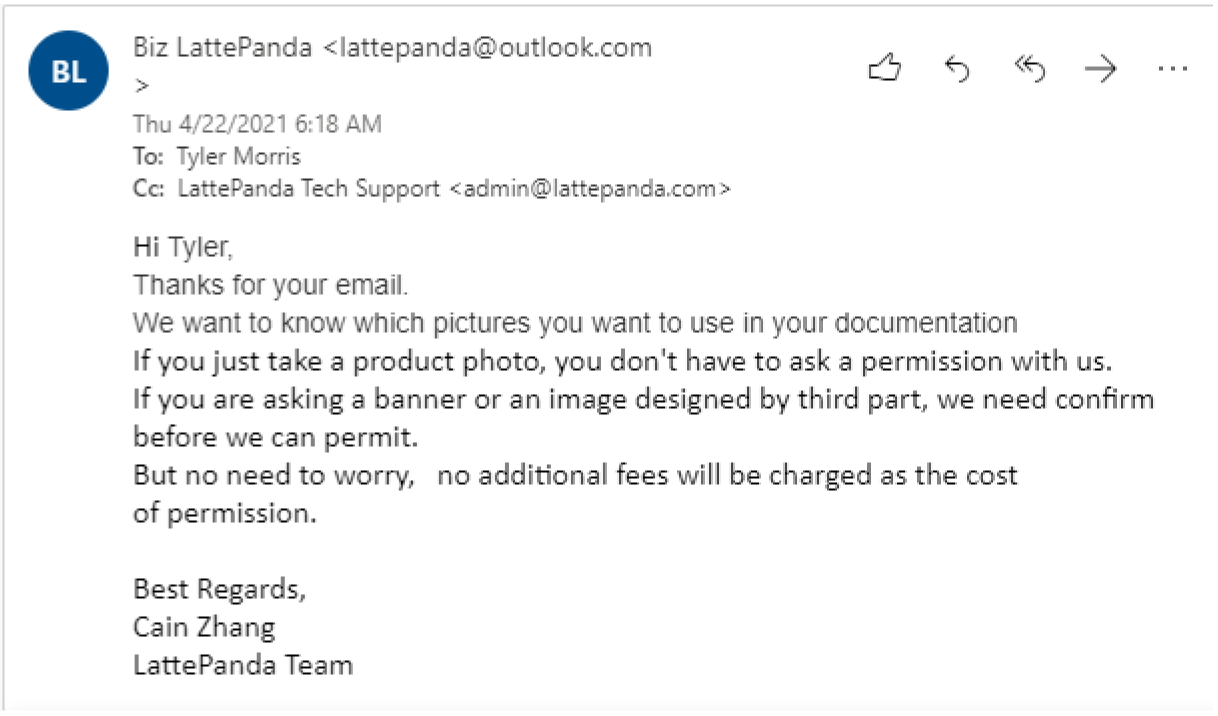
DX

Da Xue <da@libre.computer>
Mon 4/19/2021 11:59 AM
To: Tyler Morris

Hi Tyler,

You have our permission to use the image for your project work for the ECE department at UCF.

Best,
Da Xue



BL

Biz LattePanda <lattepanda@outlook.com
>

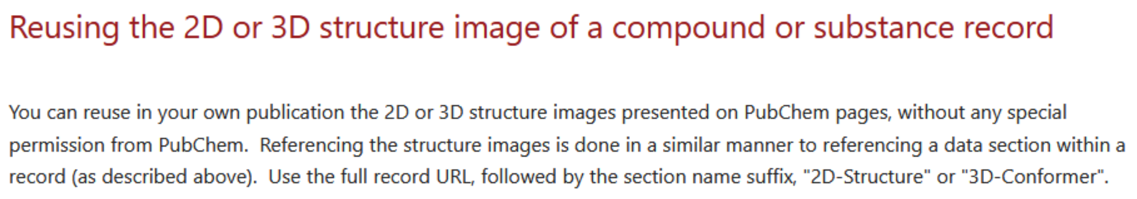
Thu 4/22/2021 6:18 AM

To: Tyler Morris

Cc: LattePanda Tech Support <admin@lattepanda.com>

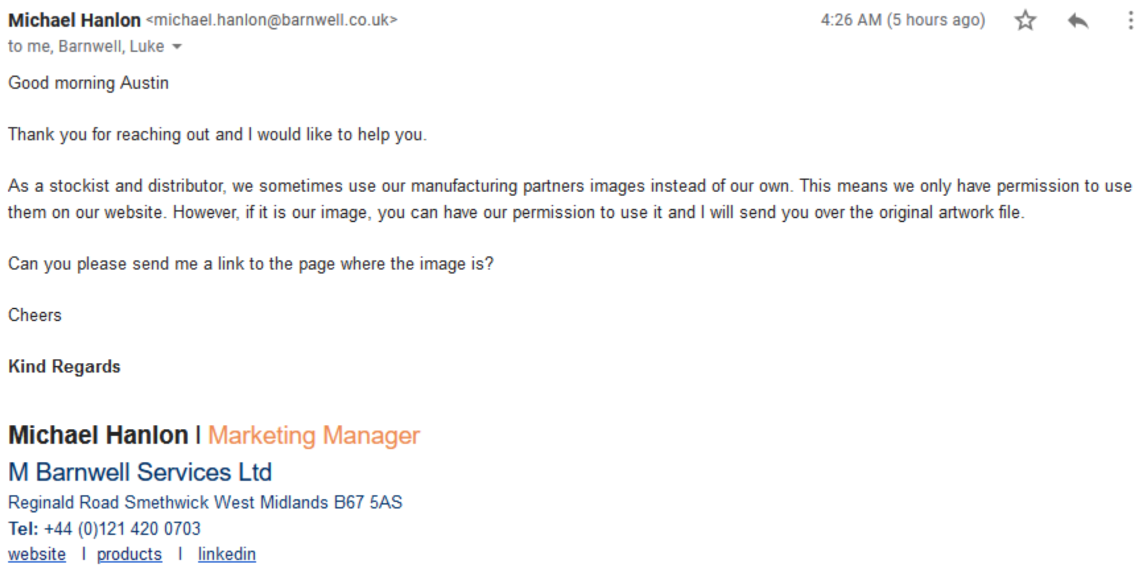
Hi Tyler,
Thanks for your email.
We want to know which pictures you want to use in your documentation
If you just take a product photo, you don't have to ask a permission with us.
If you are asking a banner or an image designed by third part, we need confirm before we can permit.
But no need to worry, no additional fees will be charged as the cost of permission.

Best Regards,
Cain Zhang
LattePanda Team



Reusing the 2D or 3D structure image of a compound or substance record

You can reuse in your own publication the 2D or 3D structure images presented on PubChem pages, without any special permission from PubChem. Referencing the structure images is done in a similar manner to referencing a data section within a record (as described above). Use the full record URL, followed by the section name suffix, "2D-Structure" or "3D-Conformer".



Michael Hanlon <michael.hanlon@barnwell.co.uk> 4:26 AM (5 hours ago)

to me, Barnwell, Luke

Good morning Austin

Thank you for reaching out and I would like to help you.

As a stockist and distributor, we sometimes use our manufacturing partners images instead of our own. This means we only have permission to use them on our website. However, if it is our image, you can have our permission to use it and I will send you over the original artwork file.

Can you please send me a link to the page where the image is?

Cheers

Kind Regards

Michael Hanlon I Marketing Manager
M Barnwell Services Ltd
Reginald Road Smethwick West Midlands B67 5AS
Tel: +44 (0)121 420 0703
website I products I linkedin