# UV Water Purification and In-Line Water Quality Monitoring with Raman Spectroscopy

Kendra Kordack, Bradley Blackburn, Grant Cooper, and Lucas Heredia

### Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

### ***Abstract* —** Having access to clean water and food are the very basic requirements for human life. Nonetheless, having a reliable and, more specifically, a clean supply of water is a requirement for the development and maintenance of any civilization. Unclean water can carry bacteria, viruses, and many harmful pollutants. Our goal was to make a system that is easily transportable, cleans water for consumption, analyzes the water in real time, and gives a report of the system and water purification.

### ***Index Terms* —** S**pectroscopy**

### I. Introduction

In most places water filtration is common and not given much thought, but in some places clean water is hard to come by. In these places, it’s expensive and difficult to install the water filtration systems commonly used. There are portable filters that can be used in these areas, but they often don’t filter out everything harmful and don’t combine filters and ultraviolet light (UV) treatment. UV water treatment can sometimes be more effective on certain bacteria that grows in water, than other water cleaning techniques. UV cleaning can also be inexpensive. When UV cleaning is combined with water filtering, such as sediment filters and carbon filters, the likelihood of consuming contaminant water is extremely low.

In addition to the UV treatment and filtering, no current transportable system contains in-line water quality monitoring. If the quality of the water being monitored, there is no way to tell if the system is working properly and the water is safe to consume. Spectroscopy is a common technique used to identify chemical compounds, and can be used to monitor contaminants in water. Raman spectroscopy is the best technique for monitoring because water is a weak Raman scatterer. Another method for monitoring water is by temperature, certain temperatures can allow bacteria to grow quickly and thrive, which would make the water harder to clean. By monitoring the temperature of the water using infrared light, we can get a fast easy reading with no contact.

In order to further optimize this system and make it ideal for places with minimal access to resources, the best option for powering the system is solar power.

In order to show the results of the monitoring a user interface is required. This can be accomplished using a microcomputer and touch screen display. The display will not only show the water quality monitoring but also show data of the solar power, time water if filtered, and battery percentage.

This prototype was sponsored by Ocean Insights, who provided the spectrometer and laser module, BCI technologies, who promn nhvided the battery and funded for this project, and MKS Newport, who provided optical components for the spectroscopy system.

### II. System Components

The system is best broken down into the individual components that are interfaced to create the entire system. This section provides an overview of each major component used in the system.

1. *Solar Power System*

The power generation for this system will be provided by two 100 Watt Polycrystalline solar panels from a company called Sun Gold Power. They will produce 12VDC to 24VDC in order to charge the battery via a 12VDC, 20A charge controller. These solar panels have the following dimensions: 36 x 26.3 x 1.1 inches and weigh about 14 pounds each. This made it hard for us to create a fully portable system, but they provide the necessary charging capabilities for the battery.

1. *Battery*

The battery being used in the system to power the various components in the system is a Universal Power Group (UPG) battery that was donated to the group, courtesy of BCI Technologies. Its specifications are as follows: model No. UB12750 and UPG No. 46082, rated for 75Ah, at a nominal voltage of 12V [Reference 1]. It is a sealed lead acid battery which offers full Depth of Discharge (DOD) for about 200 cycles. This amp-hour rating of the battery is slightly smaller than the 83Ah rating calculated in the design process, but that calculation was also an over approximation.

1. *Charge Controller*

The original charge controller that was going to be used in the system came with the solar panels as a package deal, which was a 12VDC, 20A Pulse Width Modulating (PWM) controller. After the design review process it was suggested that the group build our own charge controller or order a different one that could add more engineering complexity to the design. The new charge controller is another 12VDC, 20A charge controller from CirKits, Model: SCC3 [Reference 2]. The specifications are as follows: maximum solar charging current of 20A, nominal battery voltage of 12VDC, and night time battery current drain of 0.8-1.8mA.

1. *Control Relay System*

It was quickly realized that the RaspberryPi should not directly turn on the loads of the system, as this type of current draw would likely damage it. So incorporation of a control relay system was necessary. The RaspberryPi also only provides output signals at around 3.3V which does not have enough power to flip a 12V relay, so to boost the output signal an Opto22 G4PB8H I/O board was used [Reference 5]. This gives us the capability to output up to 60VDC at 3A if needed, but in this project 12VDC logic will be used. These output signals will control the Automation Direct 781-1C-SKT relays that will act as switches to connect each component to its corresponding PCB power supply. This System separates the small level hardware from the heavy loads of the system.

1. *Microcomputer*

The hardware chosen for this system was based on the specifications we researched for the OceanView software and the Omnidrive libraries, which were eventually decided to not be used (more on section VI Software Details). For the Main processing part we choose to use the raspberry Pi model 4B. We decided to use this small computer because of its costs and capabilities. It is the only raspberry pi that has enough RAM memory to support the OceanView software. Since Ocean View is the most resource hungry application in this project, the other applications shouldn’t have a problem. Its role in this project was to process the data it receives from the spectrometer, connected via USB cable. Since this microcomputer is popular, it had a lot of documentation available, as well as a big community providing lots of trouble shooting techniques. Another advantage of the raspberry pi, are the separated modules created to interface it with different types of hardware. The raspberry pi also has a standard touchscreen display that is an add-on to it. This display is interfaced automatically with the PI using a DSI cable.

The PI can be easily powered by a steady 5 volts direct current source. The Raspberry pi is, however, not waterproof therefore we will put it inside of a waterproof case. Because if it gets wet the whole system will be ruined.

1. *Pump*

Our team decided upon a Flojet 03526-144A Triplex Diaphragm Automatic Water System Pump [Reference 6]. There were many attractive features about this pump that made it the best fit for our design. Firstly, the Flojet is a self-priming pump and can run dry without water and not damage itself. Furthermore, this pump was intended to be used on a marine vehicle or an RV so it runs off 12V DC instead of 120V 60Hz AC. Our system is utilizing solar power which is all DC voltage, so by using this pump we will not need DC to AC conversion. It also pumps at an average rate of 2.9 GPM, at around 10 psi. This pairs nicely with our UV sanitizer system which operates at around 3.5 GPM. Although the pump has a slightly lower flow rate, this allows for a prolonged exposure to the UV light which can work to disable more resistant pathogens.

1. *Filtration System and UV Treatment*

It was concluded that the best choice of water filtration system for our project’s needs is the UV system. These systems, paired with filters, work exceptionally great in getting rid of the most important pathogens. UV filters do this in a timely manner unlike distillation and do not add any potentially harmful chemicals to the drinking water unlike chlorination. These systems can not be used independently though. Our UV system will be accompanied by a filter system which includes a sediment filter to remove larger sediment that microorganism could possibly hide behind, as well as, an activated carbon filter to remove smaller particles and remove bad odor. This will also serve to take some of the most harmful heavy metals like lead and mercury out of the water. The Viqua VT4-DSW/12 that was purchased for this project has the following features; a flow rate of about 3.5 gpm which corresponds to a 96% successful UV treatment.

1. *Spectroscopy*

The spectroscopy technique chosen is Raman spectroscopy. Raman spectroscopy is often looked at for chemical and biological analysis, due to the ability to analyze samples without adding or modifying the sample. Near-Infrared (NIR) Raman Spectroscopy was chosen for this analysis because of the possibility of fluorescence interference of biological components when doing Raman spectroscopy in water. The Raman spectrometer lent by Ocean Insight reads the NIR wavelength of 785 nm. The laser module being used with the spectrometer is a 785nm source from Ocean Insight. The entire spectroscopy system is a combination of filters, lenses, and fibers that transmit the light from the laser module, focus it on the water sample and then allow the spectrometer to read the data.

1. *Temperature Sensor*

The temperature sensor chosen was the Calex mA Output Signal Infrared Temperature Sensor from Allied Electronics. The Part number for this specific sensor is PC21MT-0 which reads the infrared light given off by an object and can read from 0 degrees celsius to 250 degrees celsius. A small schematic and deprivation is located below in Fig. 1. It has a field of view of 2:1 and a fast response time. The integration, of this piece in the system, proved to be daunting. The sensor sends a 4 - 20 mA analog current signal, the variance in current representing the temperature read. So the sensor requires an analog current to voltage signal converter and then a voltage to digital converter. So, an alternative temperature to digital sensor was implemented as backup. In case we can’t interface the original sensor reliably. The back up temperature sensor is a DHT22 Temperature and humidity sensor. This sensor sends a digital serial signal that can be interfaced using the raspberry pi GPIO pins. The sensor can be operated with both 3.3V and 5V. Enabling it to be powered directly by the raspberry pi. It can read a temperature from -30 degrees celsius to 80 degrees celsius within a 0.5 degree celsius accuracy. The manufacturer also provides a python library for reading the data.

1. *Display*

An LCD touchscreen display connects to the RaspberryPi using a DSI connector or a micro HDMI cable, making it easier to set up. Since the display is touchscreen, we do not need a mechanical pad. The display contains an integrated library that provides 3D acceleration called FKMS mode. Enabling faster graphics processing. The display resolution is 800x480 pixels RGB LED. The display is powered by a 5V pin connected directly to the pi. This display has the advantage of being operable at high and low temperatures. Its temperature range of operation is from -20 Celsius to +70 Ceslsius. So, very hot ,or cold, water cannot damage it.

1. *PCB*

The PCB boards built for the project will be responsible for powering the various components in the system.. The three PCB boards were carefully designed voltage regulators that support the specific load specifications. The first is a 12V to 12V regulator that was designed to handle up to 12A of current, it will be supplying the pump and the UV filtration system. This was later changed by having the pump being run directly off of the battery due to the high in rush current. The second is a 12V to 5V regulator designed to support 4.5A, which will be supplying the RaspberryPi, the Opto22, and the laser. Finally the last PCB is a 12V to 24V regulator designed to support a small 0.3A current used by the temperature sensor.

### III. System Configuration

To better understand the system as a whole, this section outlines how each part works together. A flowchart has also been provided below to visually summarize the operation of the system.

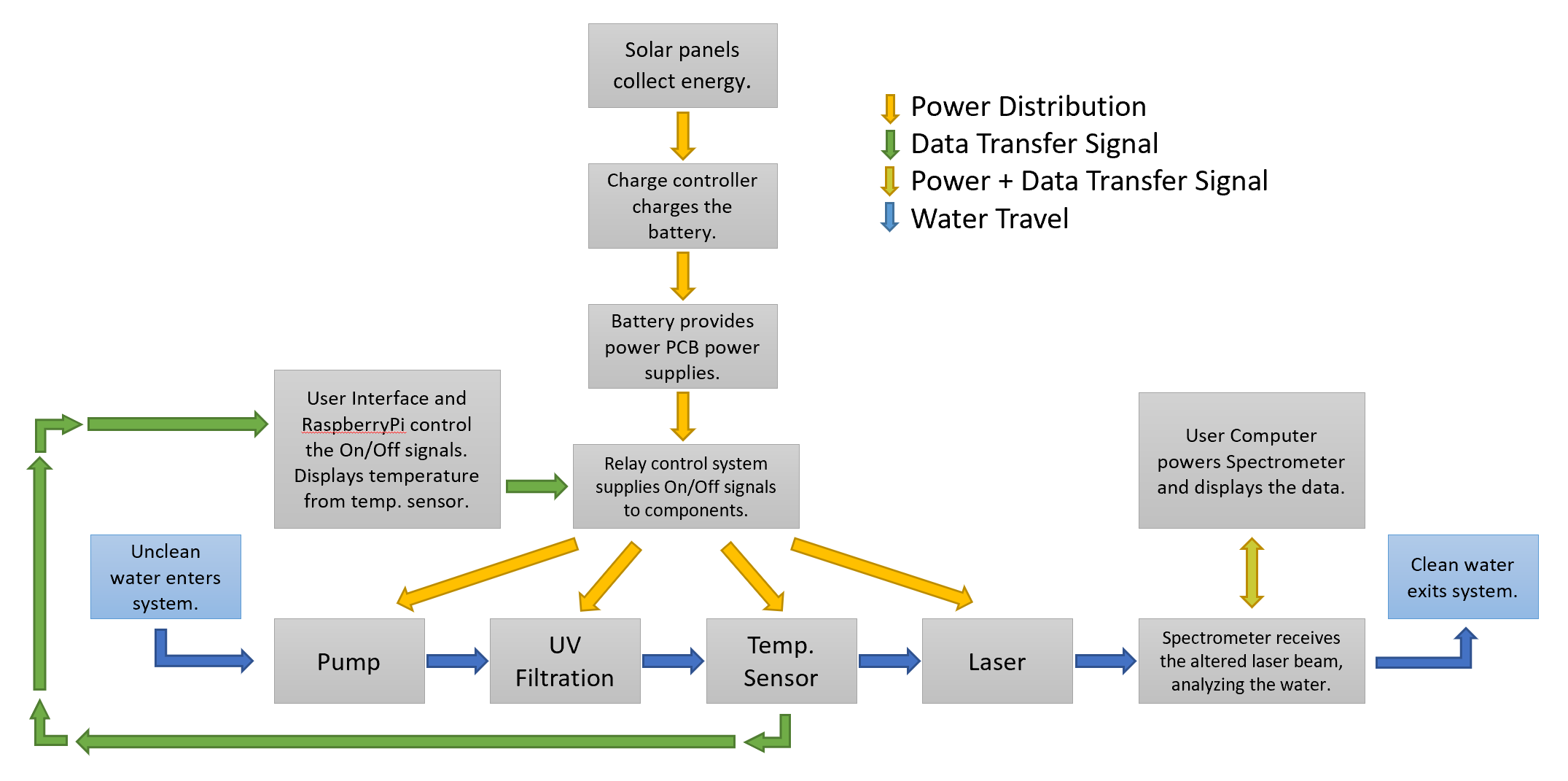


Figure 1: Flowchart of system operation

At the beginning of the system’s configuration, the solar panels convert the sun's energy into electrical power in order to charge the system’s battery. The charging of the battery will be taken care of by the charge controller, which will maintain the battery voltage at around 12V. The battery will be the overall supply for the system, powering the PCB boards which will distribute power to the various components. The three PCB boards built for the project are carefully designed voltage regulators that will allocate power to each component. The first is a 12V to 12V regulator that was designed to handle up to 12A of current, it will be supplying the pump and the UV filtration system. This was later changed by having the pump being run directly off of the battery due to the high in rush current. The second is a 12V to 5V regulator designed to support 4.5A, which will be supplying the RaspberryPi, the Opto22, and the laser. Finally the last PCB is a 12V to 24V regulator designed to support a small 0.3A current used by the temperature sensor. The spectrometer will be powered by a user’s computer via USB. This was decided once the OceanView software was found to be incompatible with the RaspberryPi and powering it off of a computer will not only interpret the data for the user, but it will also power it so the PCB load is minimized. The only true feedback that the system will see is the temperature sensor reading. The original temperature sensor gives a 4-20mA signal representing 0-250 degrees C, but the RaspberryPi does not read analog signal so an analog to digital converter module was necessary. As of 11/16/20, this is still being dealt, but a new temperature sensor has been added in and configured as a backup.

The control of the system will be taken care of by the RapsberryPi’s user interface, which will offer control over the system with virtual buttons on the LCD screen. These signals that will be generated by the RaspberryPi will be used to control 12V relays that will act as switches between the components and their respective power supply PCBs. This system is explained in further detail in the next section.

Lastly the only thing left is to add in the water. At the start of this process the pump will pull the water into the system and push it through the UV filtration system. This will get rid of most contaminants in the water. At the later end of this section, the water analysis will be done via the temperature sensor and the spectrometer. The spectrometer will provide a frequency reading of the laser passing through the water and if any contaminants show up, it will create a shift in this frequency response. LAstly, the water will exit the system and be subject to consumption but will not be at UCF does not allow person testing of experimentation. Ultimately this is the overall system concept, how each component will interact with each other and operate in the system.

IV. Hardware details

To better understand each component and their part in the system, which were outlined in Section II, the below section will explain in a more technical manner. This will include everything aside from the software and microcontroller components.

1. *Solar Power System*

As stated above in SectionII, the power generation for this system will consist of two 100 Watt Polycrystalline solar panels, one charge controller, and a sealed lead acid battery. They will produce 12VDC to 24VDC, at 100 Watts a piece. The charge controller will use the solar power to keep the battery at a specific voltage in order to keep the system powered throughout the day. Finally the battery is a 75Ah sealed lead acid battery.

To start the design process first find the overall power consumption of the system. This can be seen below where each component’s consumption was added together to get the overall load of the system, we will denote this as ‘L’ and it carries the unit of Watts.

L = 2.5 + 20 + 52.8 + 10 + 10 = 95.3 Watts

Next, we want the load expressed in Watt-Hours per day. To provide extra power capabilities, this value will be rounded up to 100 Watts. Our goal is to have 8 hours of operation per day, which we will express as ‘H’ in units of hours per day. Thus the load in units, Watt-Hours per day, denoted as Lpd is:

Lpd = L x H = 100 Watts x 8 Hours/Day = 800 Wh/day

Once again, this value will be rounded up to 1000Wh in order to provide extra capabilities. From this value we can calculate the size (in Watts) of the solar array that is needed by dividing this value by the amount of hours of sunlight that the system should receive in one day. Here in Florida there is about 5.6 average hours of sunlight per day so we will divide by hours to get our solar wattage goal. This will be denoted as ‘P.

P = 1000 Watt-Hours / 5.6 Hours = 178.6 Watts

This is how the two 100 Watt solar panels were chosen for the project. The battery Amp Hour requirement can also be calculated by taking the Watt Hour per Day value of 1000, and dividing it by 12V. We will denote this at ‘B’ in units of Amp Hours.

B = 1000 Wh / 12V = 83.33 Ah

As one can observe this is a bit higher than the actual rating of the battery, but by using this battery that was donated to the group from BCI Technologies, it was saving about $150-$200 in the budget.

1. *Control Relay System*

Since the RaspberryPi does not have the capability to run all the loads directly off of it, there needed to be some separation between the Pi and the loads via Relays. Also the Pi does not send a strong enough signal in order to control the provided relays, so another boost of the signal was needed via an Opto22 G4PB8H I/O board.

The Opto22 G4B8H I/O board will utilize 60V/3A rated output modules which will act as a switch in order to transfer a 12V control signal. This 12V signal will be brought from the 12V power supply PCB which will activate the relays, in turn connecting the components to their respective power supplies, and ultimately turning them on via the output signal from the Pi.

The relays that are being used are Automation Direct 781-1C-SKT relays which run off of a 12V coil. These relays are rated to pass a signal up to 28VDC at 15A and will act as the switch between each component and their respective power supply PCB. The wiring of the relay (visualized in the Fig. 2 below) will be as follows; the supply voltage will be connected to pin 9 of the relay, the corresponding load will be connected to pin 5 (the normally open contact), a -12V will be connected to pin 14, and the +12V control signal wire will be connected to pin 13. Between pin 13 and 14 there is a coil that, when activated, connects pins 9 and 5, which will turn on the specific component.

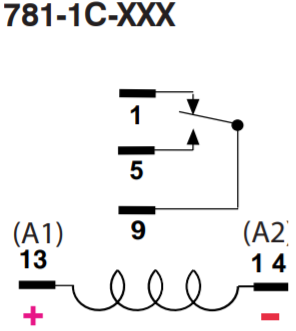


Figure 2: Pin out for the 781-1C-SKT [3]

1. *Filtration System and UV Treatment*

A critical part of our design was that it would be mobile. Most mobile devices do not use 120V AC 60Hz supplies but instead 12V DC from a rechargeable battery. Thus, both the water pump and UV filter needed to run off of 12VDC. It is much harder to find DC voltage components but lucky both devices are used in marine applications or in RVs, so we eventually found parts that met our design specification. Furthermore, when choosing our components. The highest priority characteristics was each device's Gallons Per Minute (GPM). The Viqua VT4-DSW/12 incapacitates pathogens at a rate of 3.5 gallons per minute. Accounting for a margin of error, we decided the Flojet’s 2.9 GPM would suit our design’s specifications perfectly. Figure 3 was made to depict the design of the pump and filtration system. The pump is also a pressure sensing pump, so the option to have the system run based off of just opening and closing a valve. The filtration system includes a 5 micron sediment filter, followed by a carbon filter, which leads to the UV chamber for final treatment. This should take care of most, if not all, sediments and carbon contaminants. The UV will destroy any organic matter or organism, as well as take away their means of reproducing. While the goal is not necessarily to create clean water, this will provide the spectrometer analysis a means of reference when analyzing the water.

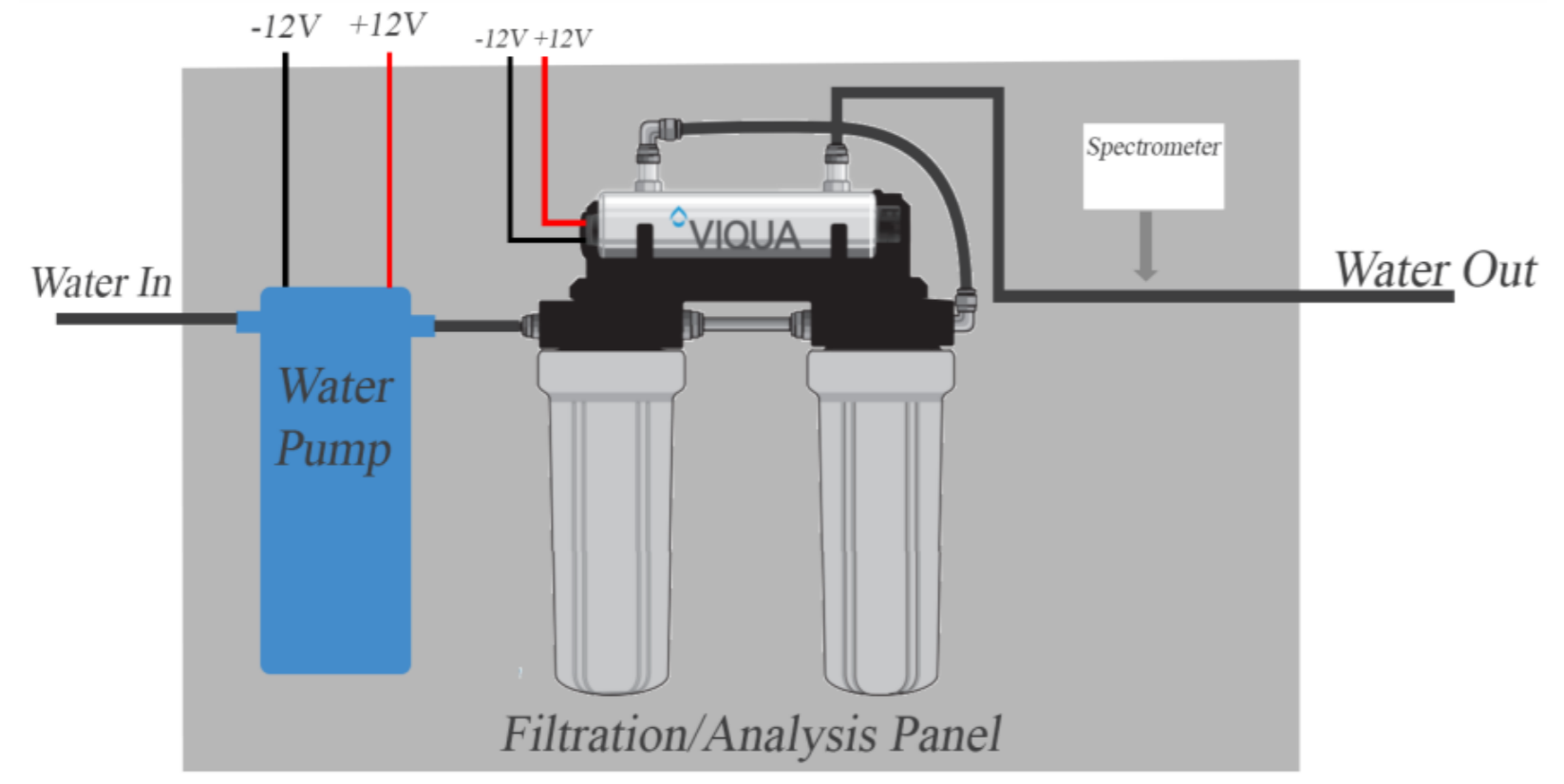


Figure 3: Pump and Filtration system design.

1. *Water Analysis*

The water analysis includes Raman spectroscopy and temperature sensing. The Raman spectroscopy system beings with a laser module supplied with 5 volts and operating at 785 nm. The laser light comes out of a FC fiber with a fiber collimator connected to the fiber. The light comes out of the collimator is 2 mm in diameter and passes through a neutral density filter to attenuate the light and a bandpass filter made for 780 nm light to clean up the incoming beam. The light then passes through a 50 mm convex lens that focuses the beam onto the water sample. After the light passes through the water sample it enters a 50 mm concave lens to collimate the beam before it enter the longpass filter. The longpass filter blocks out light not scattered by the water sample. The light passed through the longpass filter then enters an 15.29 mm NIR coated lens that will focus the light down into the SMA fiber connected to the spectrometer. A schematic of the optical setup is shown below in Figure 5.

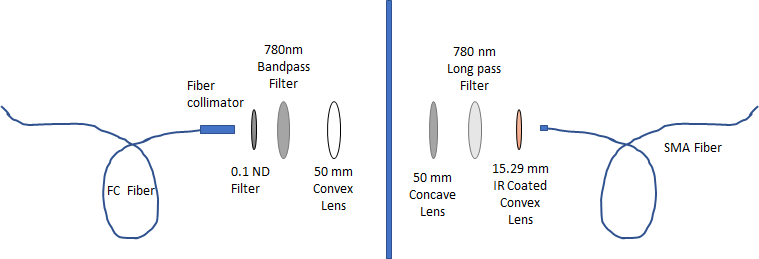


Figure 4:

For the temperature sensing is done with an IR temperature sensor that reads the infrared light the water it emitting. Since it has a 2:1 ration the sensor needs to be placed at most 200 mm from the water sampling set up. It will be placed on the optical bread, but far enough away from the laser module, so that the laser emission does not affect the IR reading.

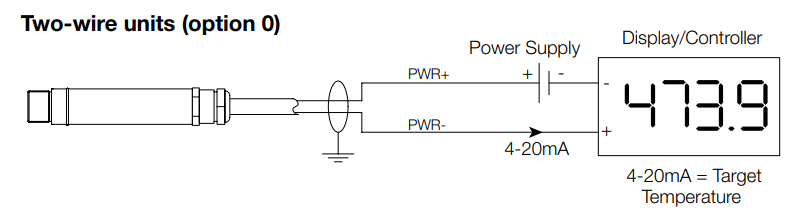


Figure 5: Wiring instruction for the IR Temp. Sensor [4]

1. *PCB*

Since our design requires three different voltage differences, three voltage regulators were produced. A 12V 12A regulator for the UV and water pump. A 5V 5A regulator for the RaspberryPi, laser, spectrometer, and touch screen. Finally, a 24V .03A regulator for the temperature sensor. A collection of figures below show each PCB in the final design software schematic. Most senior design PCBs will at most have 5As on their board but our design must be able to handle 17A. This large amperage is mostly due to the water pump. It is important to note that as the max current increases on a circuit the complexity increases with it. Connections and placements of components must be thought out and much attention must be paid toward thermal regulation. With such high currents, linear regulators were out of the question due to their poor efficiencies of at best around 60%. Rather, switching regulators were used. These regulators have much better efficiencies but introduce much noise into the circuit that must be dealt with carefully. On the 12V regulator a power ground and an analog ground were established and connected at only a few specific spots to make sure noise from power signals did not affect sensitive analog signals. Also, high di/dt currents needed to be as short as possible and placed away from analog circuitry. Assessing these complications our team decided to create the regulators on separate PCBs. This decision helped in design and allowed us to place PCBs at the point of load to avoid voltage drop over wire that could potentially render our sensitive components without power. Specifically, our laser requires a minimum of 4.9 V so much care was taken to meet this specification. Overall, due to the high current draw of our project, regulator design was quite extensive and resulted in over 60 components being placed on our PCBs. When one thinks about how every component value needs to be accurate, every trace needs to be connected exactly like the schematic, every PCB Pad and trace the correct size, how each component must be logically placed to deal with unwanted noise and heats, and how if just one out of a 1000 things are wrong, nothing will work, it’s easy to see the beauty in a working PCB.

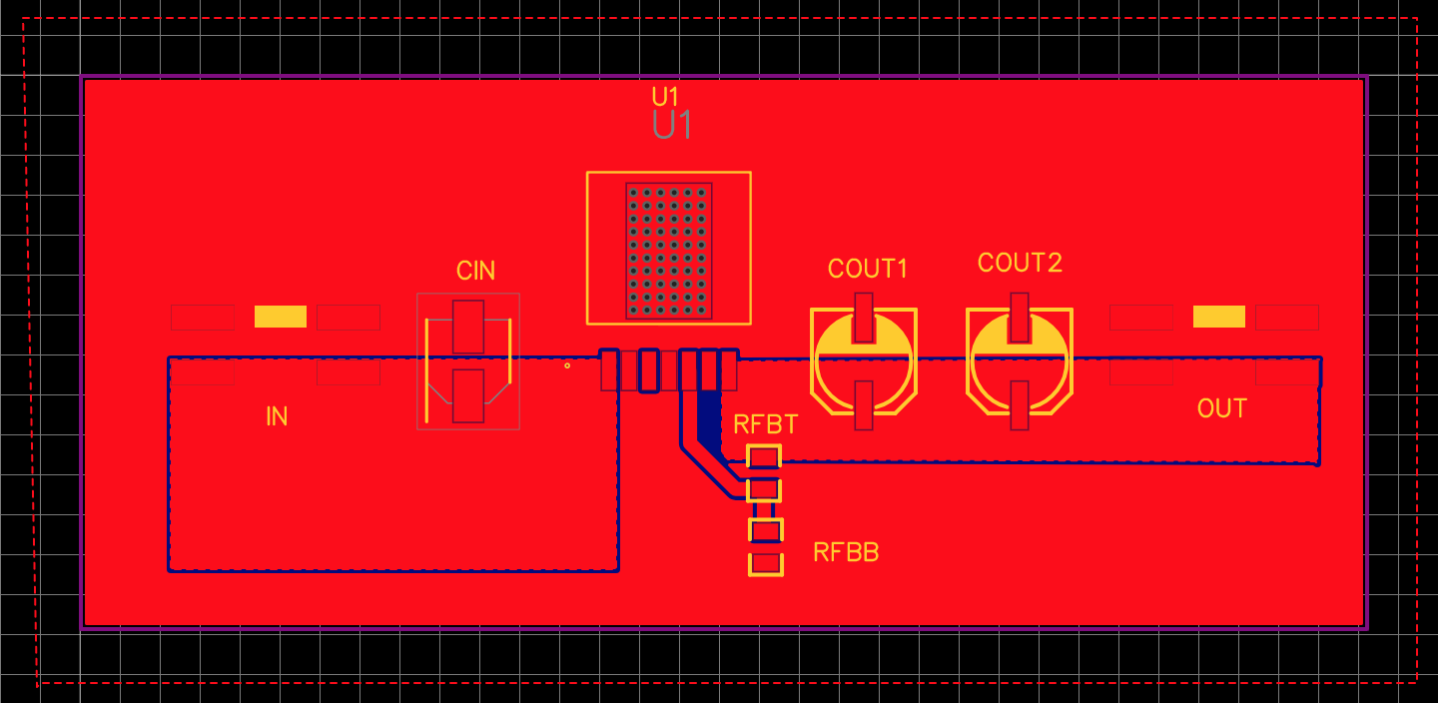


Figure 6: 5V/5A PCB schematic

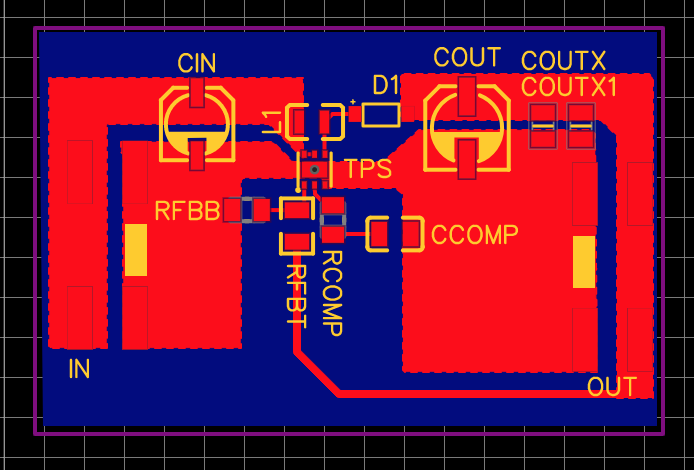


Figure 7: 24V/0.3A PCB schematic

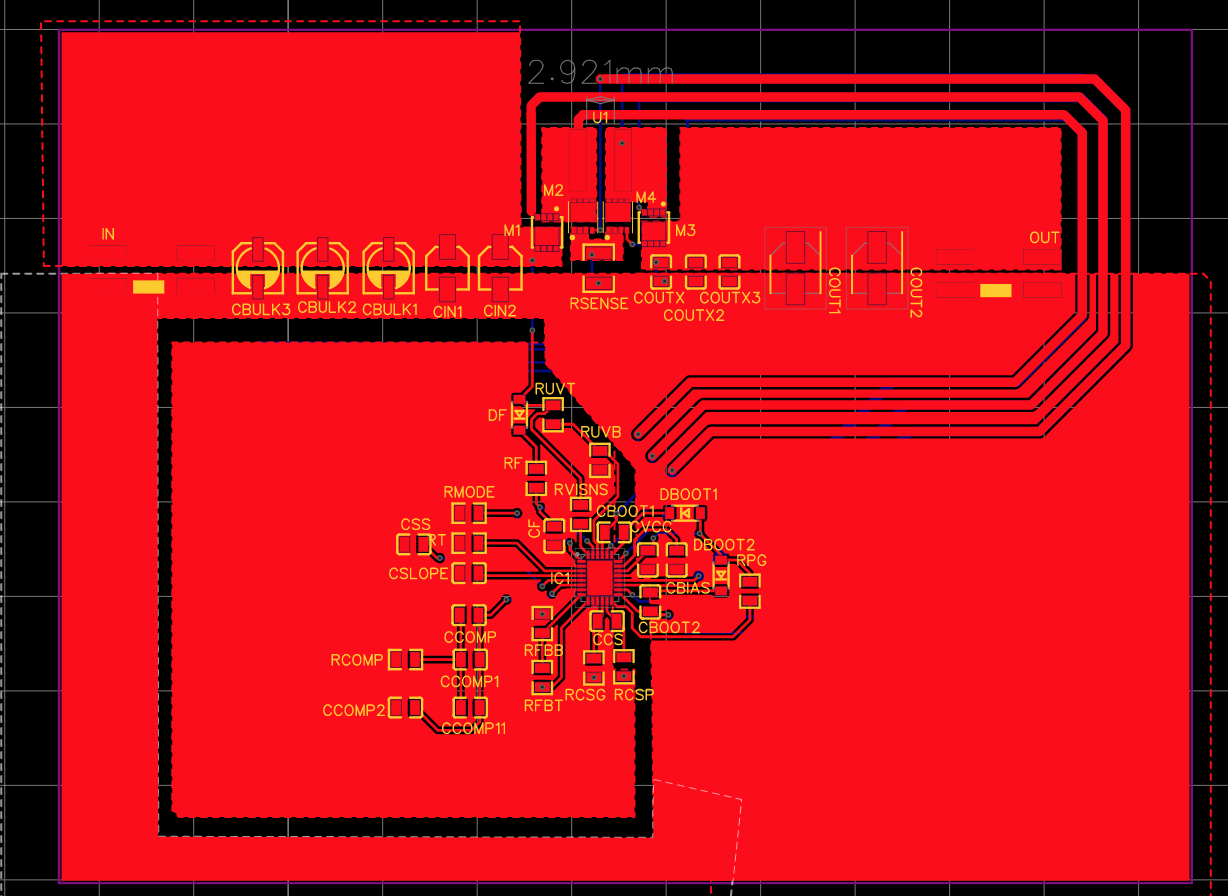


Figure 8: 12V/12A PCB schematic

### V. Software Details

To better portray the way our software system is working and set up, each part was separated in the sections below.

1. *OceanView*

OceanView is a paid program developed by Ocean Insights that enables the user to control a Ocean Insights spectrometer and process its data. Once the program is started, it scans the computer USB ports in order to interface the spectrometer. OceanView runs regression optical algorithms to average the data and display it in the form of a graph (spectra). The program also features useful tools that enable the user to reduce the noise in the data, control the light collection timeouts, filter peaks, and control other data flow components. OceanView, in its default setup collects the light for approximately 1 msec, creates an average spectra from the multiple spectra taken and shows this average spectra to the user.

This program uses a special collection of libraries Ocean Insights developed called OmniDrive. Initially, our project intended to use the OmniDrive libraries to control the spectrometer, but Ocean Insights refused to give us the key to access the OmniDrive library. The software we originally intended to design was in fact very similar to oceanView itself, except it would additionally control the components of the overall system. Since oceanView software is licensed, it doesn’t enable us to access its data using any third party software. Although OceanView allows the spectra processed to be saved, it must be done manually. Given the constraints, we decided to run Ocean View in parallel with our main program in order to process the spectrometer data. Since there is no way to interface OceanView with our application. OceanView also has compatibility issues when running in a linux system. In order to install OceanView in a linux system it requires the following libraries to be installed:

-libstd c++,

-libXp,

-libusb v.

But even though installed it still doesn’t recognize the spectrometer. The program has no issues when running on a Windows desktop OS though.

1. *Java / javaFX*

Since we originally thought we would be using the Omnidriver library to develop the application, we started developing the system using the Java programming language. Omnidriver was programmed using java. Therefore, if we also used java we would be programming natively and this could potentially decrease the amount of compatibility issues when dealing with the system. In the documentation it says that Omnidriver is also compatible with C#, Visual Basic, and C.

The first Graphic User Interface (GUI), of the system, was developed using Java FX. We decided to use javaFx because it is one of the most modern looking Java GUI libraries. JavaFX is also relatively new compared with its counterparts such as swing or AWT. We decided to use this library although javaFX support was discontinued by Oracle on Java 8 and, since then, it became open source. Now JavaFx is supported by the OpenJFX community.

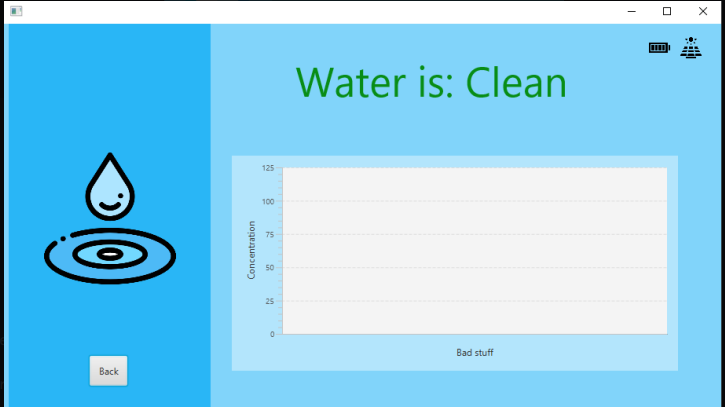


Figure 9: The original UI design

Since Ocean Insights did not provide us with the Omnidrive library, the goal of the gui became to control the components of the system: the UV light, the Pump, the temperature sensor, and the laser. These components are all connected to the raspberry pi GPIO pins.

Initially we intended to modify the original JavaFx GUI to control the raspberry pi GPIO pins. And in the process use java libraries to perform embedded programming. When dealing with this raspberry pi model, there is a popular library to interface with the GPIO pins “pi4J”. This library was made to control the raspberry pi pins using the java environment. We were, however, unable to use the library because it was outdated for this specific raspberry pi model “Raspberry 4 B” . We could not change the raspberry pi to a previous model because this raspberry pi model is the only model with enough RAM memory and processing power to run Ocean view. We were able to circumvent this problem using a database using an SQLite database and a python script. Since the raspberry GPIO library is updated for the raspberry pi.

In the end we decided to scrap the previous GUI because of another compatibility issue with JavaFx when running in the raspberry pi Linux operating system. The java virtual machine is not able to recognize the javaFx libraries in a linux system.

1. *Operating system*

The raspberry pi runs on the Ubuntu 20.04 LTS distribution. Ubuntu is a cheap, fast and light OS. It is also not licensed enabling us to modify it at will. It also possesses a big community to provide support in case we run into further issues. Most of the software we tried to use was compatible with this system. Unfortunately there were compatibility problems with the spectrometer and the GUI.

We run into less issues when using a windows operating system. but unfortunately the raspberry pi doesn’t support the full desktop Windows 10 version. It only supports “Windows 10 IoT“ and in this version nothing is compatible.

1. *New System/ Conclusion*

In the end we decided to redesign the GUI using a python library called GUIzero. Guizero is similar to java swing so it doesn’t look as modern ,but the GUI is fully functional.

The rpi.GPIO library is used to control the raspberry GPIO. This library is developed by the raspberry pi community itself and it is completely updated.

Another program we used is SQLite, for data storage. This program runs in parallel with our application to create a database and to store the program states. In order to control the backup temperature sensor we used Adafruit\_DHT, a python library for reading the serial input and it is compatible with both the temperature sensor and the backup temperature sensor.

The program initializes the GPIO ports and switches the signal every time the button is pressed thus turning its components ON or OFF. Also, the program collects the temperature in the sensor every 2 seconds and displays it in the GUI.

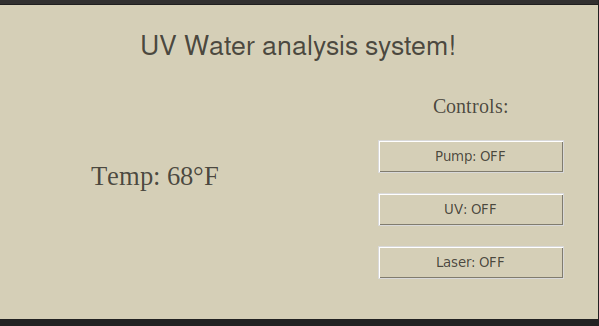


Figure 10: The home screen of the UI.

VI. Conclusion

Overall, this two-semester long project enforced working in a group with other engineering focuses. There had to be multiple comprises and understanding of each person’s part while also evenly distributing work. Multiple complications were faced, from COVID-19 restrictions to parts coming in late, that we had to overcome as a group while still maintaining our intended progress on the prototype.

For this prototype we were able to put concepts we learned from our classes and independent research on certain components of the system. Not only did we have to account for the electronics, computer science, and photonics of the system, we had to do some research on plumbing and figure out the best way to integrate all the parts together mechanically.

Furthermore, it shows that while we had all different focuses, we were able to come together as a group and combine everything we have learned while also bringing in knowledge unrelated to our specific focus to complete the project.

### Acknowledgement

The authors wish to acknowledge the assistance of the three companies that sponsored our project, Ocean Insight, BCI technologies, and MKS Newport.

### References

[1] UPG (Universal Power Group). (n.d.). Sealed Lead-Acid Battery.

Http://Files.Upgi.Com:8086/UPGFileService.Svc/GetSpecSheet/46082 .

Retrieved July 16, 2020

[2] SCC3 - 12 Volt 20 Amp Solar Charge Controller. (2009). Http://Solorb.Com/Elect/Solarcirc/Scc3/. http://solorb.com/elect/solarcirc/scc3/

[3] Automation Direct. (01/01/18). 78 Series Electromechanical Relay Selection Guide [Dataset]. Automation Direct. https://cdn.automationdirect.com/static/specs/78relays.pdf

[4]*Compact Non-Contact Temperature Sensor with Optional Dual Output*, 14 Mar. 2015, www.alliedelec.com/m/d/041b7846c30dfd91f02d93736bbaaea6.pdf.

[5] G4 DIGITAL O/I MOUNTING RACKS. (2017). Https://Documents.Opto22.Com/0274\_G4\_Dig\_I\_O\_Racks\_Header\_data\_sheet.Pdf. https://documents.opto22.com/0274\_G4\_Dig\_I\_O\_Racks\_Header\_data\_sheet.pdf

[6]“TRIPLEX DIAPHRAGM” 3426 / 3526 / 3626 SERIES AUTOMATIC WATER SYSTEM PUMP. (n.d.). Solacity. Retrieved 2020, from <https://s1.solacity.com/docs/Flojet/R3426_81000-384.pdf>

### The Engineers

**Kendra Kordick,** a senior photonic engineering student at the University of Central Florida. She is currently working at Beam Co. in optics and liquid crystal technologies.

**Grant Cooper**, a 23 year-old Electrical Engineering student. He is currently interning at a RF electrical engineering consulting firm, Athena-Tek. Grant hopes to pursue a career in RF Telecommunications engineering



**Bradley Blackburn;** Graduating with an Electrical Engineering degree and a minor in business administration from the University of Central Florida. Currently employed at BCI Technologies as a junior engineer, working on system integration. Open to all opportunities and has goals of owning a successful business.

**Lucas Carvalho Heredia,** a senior Computer engineering student at the University of Central Florida. Currently interning at Via as an application engineering intern.