Blue People – Silver Ion Point-of-Use Water Treatment Device

Kathryn Wason, Isaac Roberts, Joshua You

December 16, 2019

Capstone Design ECE 4440 / ECE4991

Signatures

/s/ Kathryn Wason

/s/ Isaac Roberts

/s/ Joshua You

Statement of work:

Kathryn Wason:

I was responsible for all of the hardware design associated with the development of the device. I created the header board we used for much of the initial testing as well as designed and created the layout for the final PCB. I also selected all of the components we used. Additionally, I facilitated communication with Mark for the development of the enclosure and with Courtney and Jim for the approval of the design. I worked with Isaac to test the device throughout the entire development process and worked with Ben and Jeff at WWW to assemble and troubleshoot the device. I also designed and developed the adapter board we used to program the devices and that will be used to retrieve the data from the prototypes at the end of the field test.

Isaac Roberts:

I designed and coded the overall embedded software structure for the system as well as many of the subsystems. These responsibilities included low and high-power loops, interfacing with the comparator, LED logic, creating an FSM to control the silver release, interfacing with the ADC10, creating systems and a memory structure for serial flash and UART, etc. Additionally, I wrote the python script used to interpret the raw data pulled from the MSP430 and tested/optimized the silver release FSM extensively.

Josh You:

I was originally in charge of the embedded software programming and the enclosure of the system. However, after some research on waterproofing enclosures and learning about additional details and specifications about the final deliverable, we determined that it was out of the scale of the project and that it would be smarter to spend our time on the electrical and computational aspects of the project. So, we decided to outsource our enclosure with the new specifications. To list the changes, originally, the system was thought to be all-in-one with the bucket that will hold the water. So, we designed the system to be a box/container that would hang on the inner wall of the bucket and have the silver rods sticking out of its side in contact with the water. However, we learned from the Civil Engineering department that these systems will actually be field tested in South Africa and that shipping 30 10 L buckets on the luggage compartment of a plane was going to be impossible, so we changed our design to be a small device that would float in the water and have the silver rods sticking out on the bottom to in contact with the water. But after that decision, the design process of the enclosure was taken over by our outsourced partner.

For the software side of the system, I was in charge of the capacitive touch sensor, UART communication, the Python data export software, flash storage, and general review of the code. When we were first deciding on the type of button to use for the system, we thought a mechanical pushbutton would be the best option, because a capacitive touch sensor would not work well with water on the surface interfering with detecting of the finger. However, after some consulting from 3W, we decided to use a capacitive touch sensor, due to it being easier and cheaper to produce and removing extra complexities that would come with waterproofing the pushbuttons. We had a lot of struggles with figuring out the software implementation of the capacitive touch sensor and went to 3W for guidance. They recommended to add a discharge cycle and a dynamically changing reference value, which helped to significantly reduce the noise and fluctuation. For the Python UART serial communication package, we were able to use the built-in UART to USB converter on the Launchpad to communicate with the MSP430 from a laptop PC. However, we had troubles when using a different model of UART to USB converter, but we figured out a way through using RealTerm to get a raw file of the readings from the flash storage through UART and then using Python to convert the raw data into a readable CSV file.

Table of Contents

Contents

Abstract 6
Background 6
Constraints 7
Design Constraints 7
Economic and Cost Constraints 7
External Standards 7
Tools Employed 8
Ethical, Social, and Economic Concerns 8
Environmental Impact & Sustainability 8
Health and Safety 9
Manufacturability & Usability 9
Ethical Issues 9
Intellectual Property Issues 9
Detailed Technical Description of Project 10
Project Time Line 19
Test Plan 20
Final Results 22
Costs 24
Future Work 24
References 24
Appendix 27

Table of Figures

Figure 1: 3.3V Regulator 10
Figure 2: Silver Release Control 11
Figure 3: Comparator A+ and ADC Control 12
Figure 4: Capacitive Touch Sensor Control 13
Figure 5: SPI Serial Flash Control 14
Figure 6: JTAG and UART Control 14
Figure 7: LED Control 15
Figure 8: Device Schematic Diagram 16
Figure 9: Adapter Board 17
Figure 10: Device Layout 17
Figure 11: Adapter Board Layout 18
Figure 12: Main PCB Assembly 18
Figure 13: Device Enclosure Assembly 19
Figure 14: Gantt Chart Schedule (Proposal) 20
Figure 15: Gantt Chart Schedule (Final) 20
Figure 16: Data Export GUI 23
Figure 17: CSV File Output 23

Table of Tables

Table 1: Silver Results from Water Test 22
Table 2: 1 Unit Parts Cost 27
Table 3: Adapter Board Parts Estimate 27
Table 4: 30 Units Cost Estimate 27
Table 5: 10,000 Units Parts Estimate 28

Abstract

We have designed a portable, floatable device that releases a fixed amount of ionic silver into 10 L of stored household water. To activate the release of silver ions, the user presses a capacitive touch button located on the bottom of the device and places the device into the water. When the device senses that it has been placed into the water, it begins to release silver ions to produce a concentration of about 50 µg/L. To release silver ions, we provide a set amount of charge to silver electrodes located on the bottom of the device by charging and discharging a capacitor to achieve a predetermined total change in voltage, which in turn oxidizes the silver wire and delivers silver into the water. We have limited the device operation using a software lock to once every twelve hours or two times per day to limit the silver release to 100 µg/L per day. We have also implemented three LED indicators that will blink in correspondence to the status of the device operation to provide the user with feedback regarding the status of the water. The system is battery powered (replaceable 9V), is controlled using an MSP430 microcontroller, and is designed to last for 6 months to 1 year.

Background

In 2011, the World Health Organization predicated that by 2015, 672 million people would still lack access to improved drinking water sources, and many hundreds of million more would not have sustainable access to safe drinking water. Although the estimate has improved from 884 million in 2008, the number is still far too high [1]. The lack of access to an improved water source leaves individuals vulnerable to the effects of waterborne diseases such as typhoid and cholera. These diseases are especially dangerous to children, as they often cause severe diarrhea, which, by some estimates, kills nearly 5,000 children worldwide each day.

The National Academy of Engineering lists providing access to clean water as one of the 14 Grand Challenges of Engineering in the 21st Century [2]. They believe that people must develop and implement technologies in order to combat the world's water problems. One such technological approach is the development of point-of-use (POU) water treatment technologies. POU water treatment technologies are an effective solution to combat waterborne illnesses and are currently employed in many households around the world. The technologies are not reliant on state or community-run water infrastructure and allow for households to treat water in their own home shortly before consumption. Ionic silver has been known to have antimicrobial properties since Roman times, and recent studies have confirmed the efficacy of ionic silver for disinfection of potable water [3]. Currently, ionic silver is employed in several point-of-use water treatment technologies, including the Folia Water paper filter and the MadiDrop+ [4][5].

However, the MadiDrop+ and other silver-based point-of-use water filtration systems do not regulate the amount of silver released into water and do not effectively oxidize the water for disinfection. As a result, the water is not properly sterilized and does not abide by the EPA secondary drinking water safety standard of 100 µg/L, which increases the risk of waterborne diseases [6]. Therefore, our team has designed an electronic point-of-use water treatment device that releases a fixed amount of silver ions into 10 liters of stored household water. The device controls the number of silver ions released using an MSP430 microcontroller and is powered by a single 9V battery. A capacitive touch sensor is located on the bottom of the device. To activate the release of silver ions, the user presses the sensor and places the device in the water. When the device senses that the user has placed the device into the water, it begins to release silver ions. To release silver ions, we provide a set amount of charge to the silver electrodes, which in turn oxidizes the silver wire and delivers silver into the water. The target concentration of ions delivered for the prototype is 50 µg/L, which is half of the EPA guideline. By limiting the device to operate once every twelve hours, or two times per day, we ensure the amount of silver released stays within the 100 µg/L standard. We have also implemented three LED indicators that will blink in correspondence to the status of the device operation to provide the user with feedback regarding the status of the water. The device is simple to use, relatively inexpensive, and power efficient.

Constraints

Design Constraints

All of the functionality required for this project is achievable with parts that are readily available from Digi-Key, Newark, and other online vendors. The only part that is integral to our design and cannot easily be swapped is the MSP430 processor. Other parts were chosen based upon both cost and power consumption considerations.

Economic and Cost Constraints

Because the goal of this project is to provide POU water filtration for extremely low-income areas, the majority of the parts used were selected with cost as one of the primary concerns. However, as this project is not designed to be disposable, the cost of continually buying batteries to power the POU system will be a factor; accordingly, the power consumption of certain components may force us to opt for more expensive and efficient options. The biggest anticipated budget item for the project is the enclosures.

External Standards

Since the device will be submerged or in contact with water for a prolonged period of time, it is vital that our device remains waterproof throughout the duration of use. In addition to the waterproofing requirements, the final product will be used in rough environments and circumstances, thus it must be able to withstand constant abuse for the expected life of the product. Therefore, our enclosure must meet the NEMA (National Electrical Manufacturers Association) and IP (International Protection Marking) safety standards, which are the North American and International standardized classification and ratings for mechanical casings and electrical enclosures against intrusion, dust, accidental contact, and water. We must meet both standards, as our product will be manufactured in the United States and sold in South Africa. The exact ratings that we need to meet are the 6P rating for NEMA and IP68 rating for IP, which means that the enclosure is protected from total dust ingress and long-term immersion up to a specified pressure, which is 1.5 meter for a minimum duration of thirty minutes [7][8][9].

The final product should also meet the US EPA Secondary Maximum Contaminant Level (SMCL) for silver, which is 0.1 mg/L. This mirrors the World Health Organization (WHO) guideline of 0.1 mg/L. Although the secondary standard for silver is non-enforceable, over consumption of silver can cause aesthetic discolorations of the skin or argyria [10].

We must also ensure our device is RoHS compliant in order to eliminate the possible contamination of water to lead. According to the WHO, soldered connections can cause intoxication in children, and the EPA states that the maximum contaminant level goal for lead is zero. This means that, “based on the best available science which shows there is no safe level of exposure to lead” [11]. Thus, we must be sure to use lead free solder and lead free PCBs.

Additionally, we have to meet the IEC (International Electrotechnical Commission) standards for electrical safety. The specific code that we should meet is the IEC 60950 Certification, which is the electrical product safety standard for mains-powered or battery-powered information technology equipment [12].

Finally, we must ensure our PCB follows IPC standards. The main IPC standard with which we complied was IPC-A-610G, which is the acceptability standard for PCBs [13].

Tools Employed

We decided to use KiCad for the schematic and PCB design. KiCad is an open source electronic design automation software that offers a high level of functionality and has no licensing fee. Therefore, if we were to pass our work off to other designers in the future, they would easily be able to obtain the software and work with our design. Digi-Key also offers a KiCad library comprised of thousands of symbols and footprints for their components which eases the pain of having to design new symbols and footprints for components not already part of KiCad that we needed for our design.

For the software programming of the MSP430, we decided to use Code Composer Studio since we all had prior experience working with Code Composer Studio.

The National Instruments Virtual Bench was used as a power source and for measuring waveforms. We chose to use it due to its availability within the Electrical and Computer Engineering department at UVa and because all group members have used Virtual Benches in previous classes.

In order to communicate with our device using USB/UART we used Realterm. We chose this program because it was free, easy to use, and had a feature for writing any data it received directly to a text file. In order to interpret the memory data written by Realterm, we used a Python script to convert the hex data back into a human readable format. Python was chosen simply because of its ease of use and versatility.

Ethical, Social, and Economic Concerns

Environmental Impact & Sustainability

The environmental impact of this device will be relatively low. The biggest affects will be from the manufacture of the 9V batteries and the other components. Since the device is designed to be reused, the only waste produced from continued use should be the 9V battery, but the low power design should limit the impact of batter disposal.

Health and Safety

The primary purpose of our device is to provide people with improved drinking water. Therefore, health and safety are our main concerns. In order to avoid possible lead contamination, we must use RoHS compliant solder and PCBs. We also must ensure we release the proper amount of silver into the water in order to comply with the EPA SMCL and WHO guideline for silver.

Manufacturability & Usability

The POU prototype is designed to purify 10L of water. The design must be robust enough such that the electronic components of the design are able to survive accidental drops, bumps, splashes, and brief submersions. The user interface must be easily understood such that it does not rely on literacy. The primary UI will consist of colored LEDs that blink in correspondence to the status of the device operation and quality of water.

Ethical Issues

The biggest ethical concern with this project will be how and to whom the water purification system is distributed. The system has the potential to be the difference between life and death for its users, so depriving an individual or family access to it could have serious implications.

Intellectual Property Issues

We believe our device is patentable. Many commercial systems exist that electrolytically release silver into water; however, our unique contribution is to make it for household use. We also present a new way of measuring the amount of silver release into the water that allows the device to release the same amount of silver regardless of the concentration. Many of the smaller features we implemented are also non-obvious and novel improvements to existing technologies.

US3923632A - Method of and apparatus for disinfecting liquids by anodic oxidation with a silver anode

This invention provides a method for disinfecting liquids, especially water-containing liquids, in which the liquid to be disinfected is subjected to the effect of an electric current in an electrolytic cell comprising an anode and a cathode; the anode being made of silver or a silver-containing material, the disinfection taking place in the electrolytic cell [14].

Like this device, our device has an anode and cathode made of silver used for the disinfection of water. However, our device does not have separate anode and cathode compartments.

US4337136A - Device for purifying water

A pair of electrodes formed of silver-copper alloy depend from the bottom wall of a floating container. The latter is arranged to float on the surface of a body of water to be treated and contains a battery which is connected in circuit with the electrodes through a timer switch and a current reversing switch. Passage of a direct current across the electrodes ionizes the water and the silver and copper ions destroy and prevent the growth of bacteria and algae [15].

Like this device, our device floats on the surface of the water, is powered from a battery, and passes a direct current across the electrodes to ionize the water and silver ions. Unlike this device, we do not use a timer switch or a current reversing switch.

US5614067A - Leaching device for electrolyzed silver
The patent claims a leaching device for electrolyzed silver comprising a pair of silver electrodes which form an anode and a cathode, an electrolysis power source for applying a DC electrolyzing voltage between both of the electrodes, a power source control circuit for ON/OFF control of the electrolysis power source, a current control circuit for controlling the electrolyzing current flowing between both of the electrodes thereby controlling the leaching amount of silver, a driving circuit for driving the polarity switching circuit, and an abnormality sensing circuit that senses the abnormality of the silver electrode based on the change of an electric current flowing to the silver electrode [16].
Like this device, our device contains an anode and cathode made from silver and a DC power source. Unlike this device, we do not change the polarity of the current or sense the abnormality of the silver electrode. Rather, we measure the amount of silver release by calculating a total change in voltage, which corresponds to a set amount of silver.

Detailed Technical Description of Project

We developed a point-of-use water treatment device that releases silver ions into water as a form of sterilization. Our device is powered from a single 9V battery and is controlled using the MSP430G2553 low-power microcontroller [17].

Our device operates from a single 9V replaceable battery. However, since the MSP430, D1213 TVS diodes, and the SPI serial flash memory chip all operate from 3.3V, we used an LT3009 3.3V regulator [18]. The LT3009 is a low dropout linear regulator with a fixed output voltage and has an ultralow quiescent current.

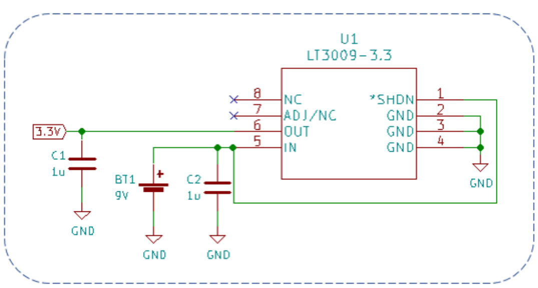


Figure 1: 3.3V Regulator

The LTC4412 creates a near ideal diode function with very low loss and low quiescent current [19]. The microcontroller GPIO pin P2.0 commands the LTC4412 through the CTL input. With a logical low input on the CTL pin, the battery supplies power to the load. When the CTL is switched high, the voltage to the capacitor is cut off. We use back-to-back ZVP3306F P-channel MOSFETs so that the drain-source diode will not power the load when the MOSFET is turned off [20]. We added a 47k pull up resistor to the CTL line to ensure that the LTC4412 does not hang between states. Using the LTC4412, we charge and discharge the capacitor to release a set amount of charge to the electrodes and release a set number of Coulombs of silver into the water.

For the capacitor, we chose a 0.47 μF film capacitor with low ESR and ESL [21]. It was important to choose a capacitor with low loss and low hysteresis to ensure optimal functionality from the capacitor.

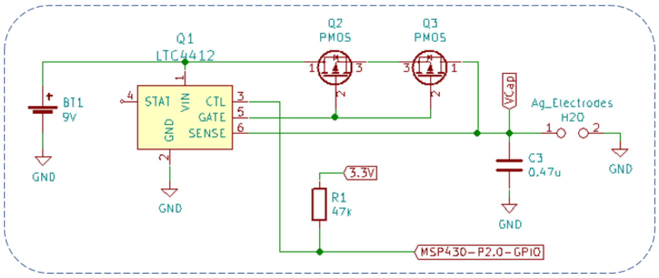


Figure 2: Silver Release Control

Since the MSP430 pins have a maximum voltage supply of 3.6V, we implemented a voltage divider to reduce the voltage seen on the Comparator A+ and ADC10 pins. In order to create the voltage divider, we used an LMC6482 CMOS dual rail-to-rail input and output operational amplifier [22]. The voltage on the capacitor charges to a maximum value of 9V, and we decided to allow the capacitor to discharge to a value of 7V. The MSP430 has two set internal references for the comparator: 0.5*VCC and 0.25*VCC. We decided to use 0.5*VCC, meaning the comparator was set to 1.65V. Therefore, we selected values for the voltage divider such that 7V seen on the capacitor would correspond to 1.65V seen on the Comparator A+ input pin P1.0.

In order to control the amount of charge released into the water by the device, we used a three-state finite state machine (FSM); we will call the states LowPowerMode, DischargeCap, and ChargeCap. The system spends most of its time in the LowPowerMode state. While in LowPowerMode, the comparator and analogue to digital converter are off, and the control to the diode is turned on (3.3 V). After receiving a valid button press (time since last silver release is greater than 12 hours), the FSM will transition from LowPowerMode to ChargeCap. On this transition, the diode control will be set to low, the comparator is turned on, and TimerA1 (TA1) is started.

TimerA1 runs at a 2MHz frequency in count-up mode. It is sourced from the SMCLK with a division of 8. Configured to count up to TA1CCR0, TA1 will trigger an interrupt upon reaching TA1CCR0 at which point the capacitor will be charged. The MSP's built in ADC is used to capture and record the voltage of the capacitor in the charged state. After the conversion in the ADC is complete, the diode control pin on the MSP is set to low (0 V) causing the capacitor to discharge.

The next state transition will be triggered by the comparator which is configured to trigger an interrupt when the voltage on its positive terminal changes from higher than its internal reference (Vcc/2 = 1.65 V) to lower than its internal reference. Within this interrupt, the ADC will again be used to measure the voltage on the capacitor. After the completion of this conversion, the discharged capacitor voltage value will be subtracted from the last stored charged capacitor voltage value. The difference between those two values will then be added to a 64-bit long long which keeps track of the total change in voltage. The FSM will then either charge the capacitor and transition back to the ChargeCap state or let the capacitor discharge to 0, clear the total change in voltage variable, and transition back to LowPowerMode. The transition out of DischargeCap is governed by whether the total change in voltage is less than or greater than a total voltage difference calculated in the preprocessor for the 10L of water the device is designed to be used on. If the voltage is greater than the target value, that FSM will transition to LowPowerMode, the total change in voltage counter will be reset to zero, the comparator will be turned off, the diode control pin will be set to low (0 Volts), a timestamp will be recorded in flash memory, a software lock prohibiting the device from releasing silver for 12 hours will be initiated, and a command to reconfigure TA1 for capacitive touch will be placed into the instruction queue. If the total change in voltage is less than the target value when the comparator interrupt is triggered, the FSM will simply transition back into the ChargeCap state.

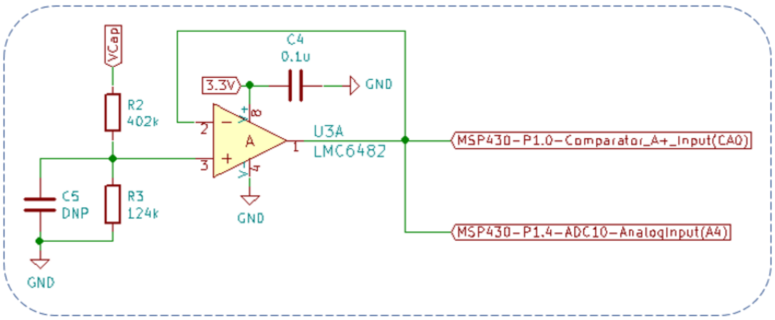


Figure 3: Comparator A+ and ADC Control

In order to start the operation of the device, we implemented a capacitive touch sensor. We chose to use a capacitive touch sensor rather than a pushbutton to avoid having to drill into the enclosure and risk water leaking into the container due to hardware complications.

The hardware of the capacitive touch sensor is a large resistor (5.1 MOhms) and a thin sheet of metal, in our case copper, which serves as a capacitor. The idea is based on the fact that the capacitance of the copper sheet will change when a human finger is in contact with the copper sheet. The MSP430 will then read the difference and determine if the difference is significant that it registers as a valid touch. To determine the change in capacitance, we can simply measure the discharge time of the circuit. We choose to use 5.1 MOhms for the resistor value, because the resistor value needs to be relatively large in order to provide any realistically measurable discharge time. The capacitive touch sensor uses two GPIO pins to charge and discharge the RC circuit.

The logic of the system is as follows. Initially, the GPIO pin P2.2 is set as an output and then set to LOW. This is to discharge the pin just in case it has not been discharged before. The pin is then set as an input, because the pin is going to be used to read/measure the discharge time. Then, the pin is configured so that an interrupt will occur on the next LOW to HIGH transition. Now, the GPIO pin P2.1 will be set as an output and then set to HIGH to charge the copper plate.

TimerA1 will be initialized, cleared, and started as soon as pin P2.1 is set to HIGH. This will eventually trigger the interrupt on pin P2.2, where the program captures the TAR register value, which will contain the count value of TimerA1 since pin P2.1 was set HIGH, and resets TimerA1. This TAR register value represents the time it took to charge the copper plate. We can then determine if a finger was pressed by comparing this value to a threshold value. Since touching the copper plate is essentially adding a larger capacitor in parallel to the copper plate, the overall capacitance of the circuit will increase by a significant amount, resulting in a longer charge time. Therefore, if the discharge time is greater than the known threshold value, we know that a valid pressed occurred. This implementation of touch sensing is vulnerable to noise and fluctuation depending on the environment, such as other electronics and humidity, so we implemented a discharge cycle in addition to the charge cycle. By summing the discharge time of the charge cycle and the discharge cycle and implementing a dynamically changing threshold value, we were able to reduce the noise to a point where we are confident in the accuracy of capacitive touch sensor in all circumstances.

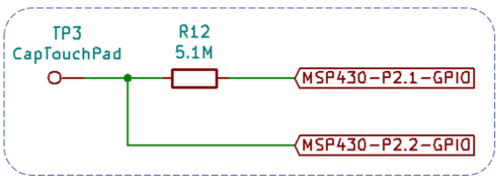


Figure 4: Capacitive Touch Sensor Control

In order to create a log of timestamps corresponding to the button presses, we used the SST25VF512A 512Kbit SPI Serial Flash chip [23]. We defined a memory structure to store in the SPI Serial Flash chip that was 18 bytes long. Each memory entry contained a 2-byte int which would correspond to an enumeration of the major functions within the system. Following the 2-byte command int, are two 8 byte values of type long long. The first of these long longs contains the timestamp of when the recorded event happened, and the second currently contains no data. It was added in the event more system information needed to be associated with each memory entry.

Accurate timestamp data was achieved by sourcing TimerA0 from an external 32-kHz watch crystal with a regular interrupt timed at every $1/8^{th}$ a second. On each interrupt, a global variable of type long long is always incremented. This long long value is used as the timestamp for any memory entries written to the SPI serial flash chip. After retrieving data from the serial flash chip, the timestamp for any entry can simply be divided by 8 to determine the total number of second that have elapsed since the device was turned on (assuming no power interruptions).

In order to optimize the power usage of the MSP430, we implemented a first in first out (FIFO) queue in order to efficiently execute any computationally intensive functions that we determined needed to be called within the $1/8^{th}$ a second interrupt. Some of these included modifying structs to flash LEDs, writing to flash memory, initializing silver release, and UART communication. At the end of every TIMERA0 interrupt, the queue would be polled. If the queue was empty, the MSP430 would be put into LPM3, otherwise it would go into active mode upon the servicing of the interrupt. When in active mode, the MSP simply enters a while loop which holds the device until the FIFO queue has been emptied and all functions have been performed, the device then returns to LPM3 where more functions will eventually be added to the queue.

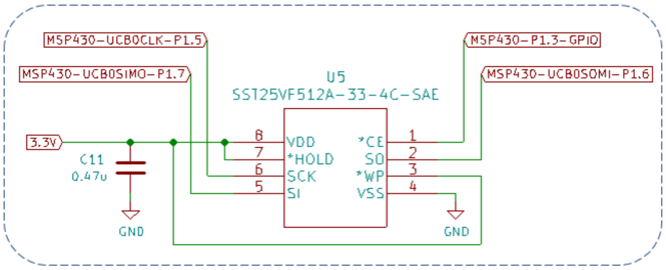


Figure 5: SPI Serial Flash Control

We used Spy Bi-Wire 2 wire JTAG to program the MSP430 microcontroller on the PCB.

In order to protect the RX/TX lines of the MSP430's UART, we used two D1213 TVS diodes, one for the RX line and one for the TX line [24]. Should a statically charged person come in contact with the pins, the TVS diodes will divert the energy to ground and away from the MSP430.

Interfacing with the onboard serial flash memory uses a UART connection with a baud rate of 9600, 8 bits, no parity, 1 stop bit, and no flow control. On the computer side, we were able to interface with the MSP using Realterm and both send and receive data. On the MSP430, we used the universal serial communication interface (USCI) modules to support UART mode. On the USCIAB0RX_VECTOR ISR, newly received bytes were added to a char array and an int tracking the number of entries in the buffer was incremented. On every interrupt from TimerA0 (every 1/8th a second), this buffer was checked for new entries. When new bytes are received, the most recent three bytes are compared to several command sequences included sequences for sending all memory entries from the onboard flash memory and for erasing the onboard flash memory.

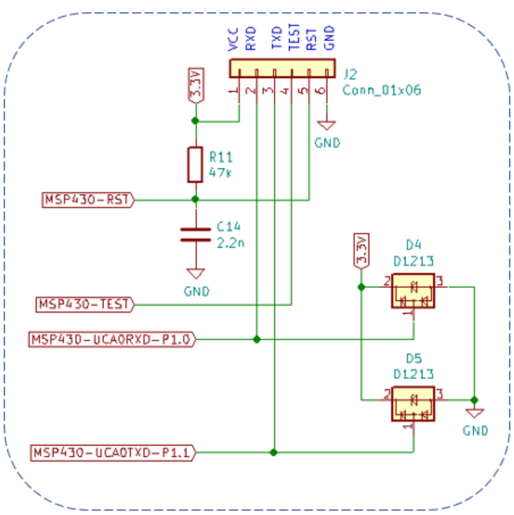


Figure 6: JTAG and UART Control

In order to provide feedback to the user to indicate both the status of the device operation and the water, we used three LEDs – a red LED, a yellow LED, and a green LED – and controlled them with DMN3135LVT dual N-channel MOSFETs [25]. We added a SD1206S040S2R Schottky Barrier Rectifier Diode at the input of the LEDs to prevent current from flowing in the wrong direction [26]. In order to control the blinking of the LEDs, we created a struct to represent LEDs. Each struct consists of 5 values which represented the struct's assigned color, remaining blinks, interval counter, interval reset value, and LED on interval value. With this structure in place, we were able to easily set LEDs to blink for varying amounts of time and at varying frequencies and duty cycles. Within the 1/8th second interrupt on TA0, each LED struct is always checked for remaining blinks, if that number is greater than zero, the interval counter is always decremented. If the interval counter is less than or equal to the LED on interval value, the LED will turn on. When the interval counter reaches zero, the interval counter is reset to the start value and the LED is turned off.

4 quick flashes from the red LED are used to communicate to the user that they have tried to initiate a silver release while the software lock is still engaged (within 12 hours of the last silver release).

Beginning after the user initiates a silver release, the yellow LED begins flashing once every 8 seconds to indicate that the water is not yet safe to drink, but is undergoing treatment.

After 4.5 hours, the green LED begins flashing once every 8 seconds to indicate that the water is safe to drink. The green LED will continue to flash until the software lock is lifted 12 hours after the silver release process was initiated.

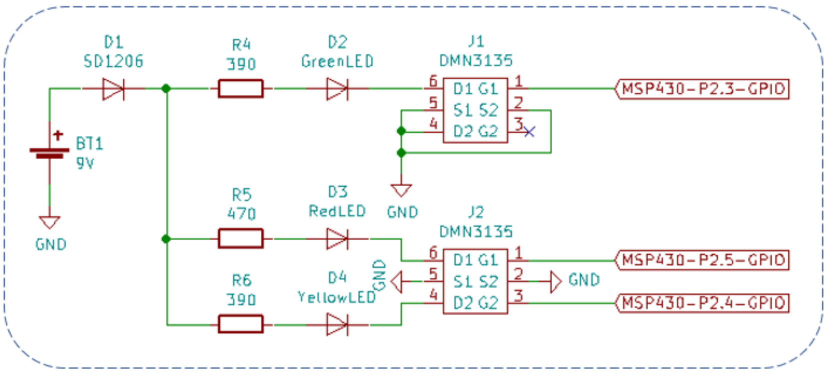


Figure 7: LED Control

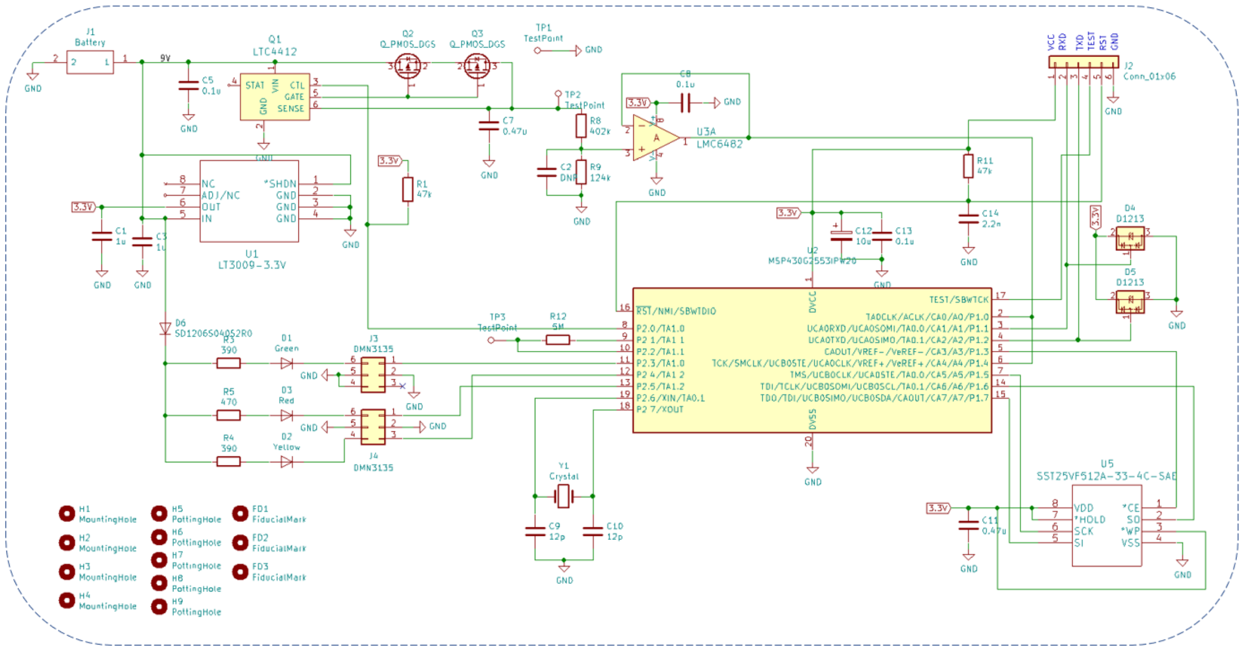


Figure 8: Device Schematic Diagram

In order to program the devices and retrieve the timestamp data at the end of the field test, we created an adapter board that includes the JTAG Spy-Bi-Wire connections as well as the USB to UART bridge converter. We chose to put the JTAG connection on the adapter board rather than each individual device to eliminate the need for an additional header on the device PCB. This saves us space on the main PCB and ensures the PCB fits within the enclosure. In order to connect the adapter board to the main PCB, we used a 6 position flat flex cable receptacle with latch to solder tab. We then placed the 6 position mating socket on the main PCB. This prevents people from accidentally connecting the header backwards and frying the components.

For the USB to UART bridge converter, we used a uUSB-PA5-ii USB 2.0 to UART interface evaluation board based on the SiLabs CP2104 USB to Serial Bridge IC from Silicon Labs [27]. The converter allows us to retrieve the data stored in the SPI serial flash memory chip.

In order to retrieve data from the serial flash memory trip, we use Realterm to send a command to the MSP430 over UART. Upon receiving the command for a memory dump, the MSP will begin sending bytes sequentially from the flash memory until it detects the first invalid memory entry (writable flash memory with all bits set to high). The MSP430 will then intentionally send an invalid memory entry (18 consecutive bytes of 0xFF) in order to signal to a python script interpreting the data that there are no more valid memory entries. The data that Realterm receives (formatted in Hex) can either be copied and pasted or directly written to a .txt file. We then use a python script to interpret the bytes based on our 18 byte memory structure and, based on a user defined initialization time, calculate when each memory event occurred based on the timestamp encoded in each memory entry. We then output the data in an easy to read excel file which decodes the enumerated command types and turns the timestamps from a count of 1/8th second interrupts into a date and time based upon the user provided date/time the device was powered on. Additionally, we included a command for the flash memory of the devices to be wiped in case the data collection process is restarted.

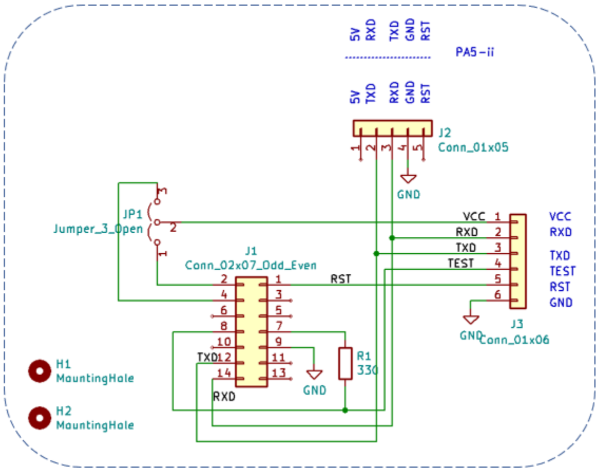


Figure 9: Adapter Board

Since we contracted with MacLean-Blevins & Associates for the enclosure design, we were given dimensions for the main PCB board outline, mounting holes, LEDs, silver electrodes, capacitive touch sensor pad, and battery holder. We created and imported a DXF file from SolidWorks into KiCad and based our board layout design on the given dimensions. We grouped components by functionality blocks and focused on minimizing trace length as much as possible. We also placed a ground plane on the bottom copper layer being sure to avoid placing it over the capacitive touch sensor pad. Due to the copper plane on the bottom layer, we routed the components with traces on the front copper layer, using vias and short traces on the bottom copper layer only when absolutely necessary. This ensured that the ground plane was unobstructed by any long traces. We also added non-plated through holes around the board to serve as air holes when the PCB is potted into the enclosure.

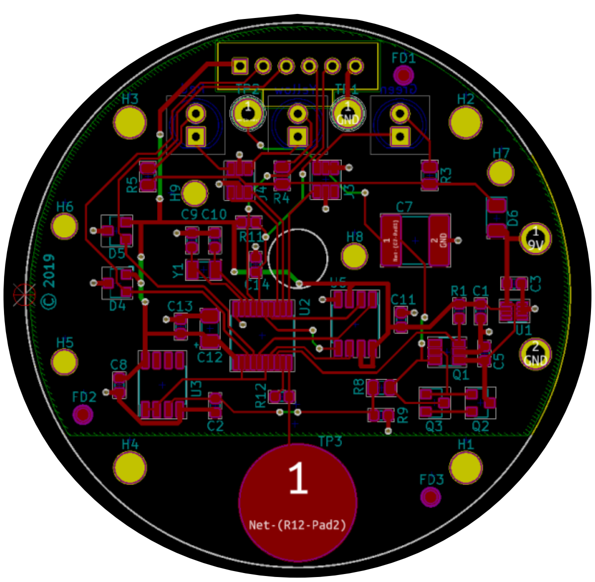


Figure 10: Device Layout

We designed the adapter board such that the user could not accidentally connect to the headers backwards and fry the components. To do so, we used a 6 position flat flex cable receptacle with latch to solder tab that is used to connect to the 6 position mating socket on the main PCB. We also used a keyed 14 pin header to prevent the user from connecting the JTAG programmer the wrong direction. Additionally, we secured the USB to UART converter to the adapter board using a 5 position right angle connector and zip tied the mini USB cable to the board so that all the user has to do is plug the USB cable into his or her computer to begin retrieving data.

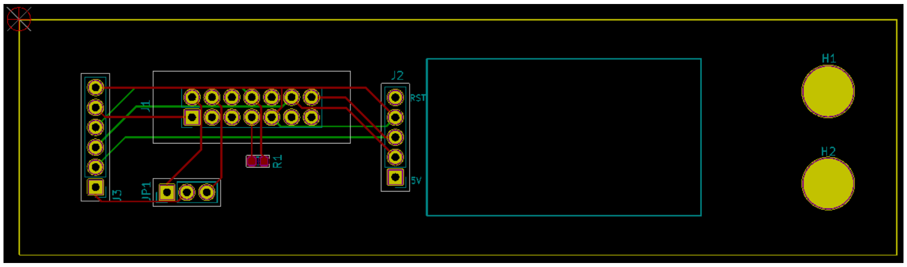


Figure 11: Adapter Board Layout

For the development of the enclosure, we worked with Mark MacLean-Blevins, who prepared the specifications and files for fabrication of the units. The enclosure includes five custom parts: the housing bottom, the housing top, the O-ring clamp, the O-ring clamp insulator, and the electrode cover. The housing bottom, O-ring clamp insulator, and electrode cover were all CNC fabricated in ABS plastic with no primer or paint, and the O-ring clamp was made in 6061-T6 Aluminum Alloy with no painting or finish treatments. For the housing top, we had to decide between using a PMMA (acrylic) or a PC (Lexan). We decided to CNC fabricate in a translucent PC, with a frosted inner surface to assist with diffusing the LED illumination. The PC provides better impact resistance, which is a more suitable choice for our application. The PMMA would polish up better, but as the same per unit cost, the PC was a better choice.

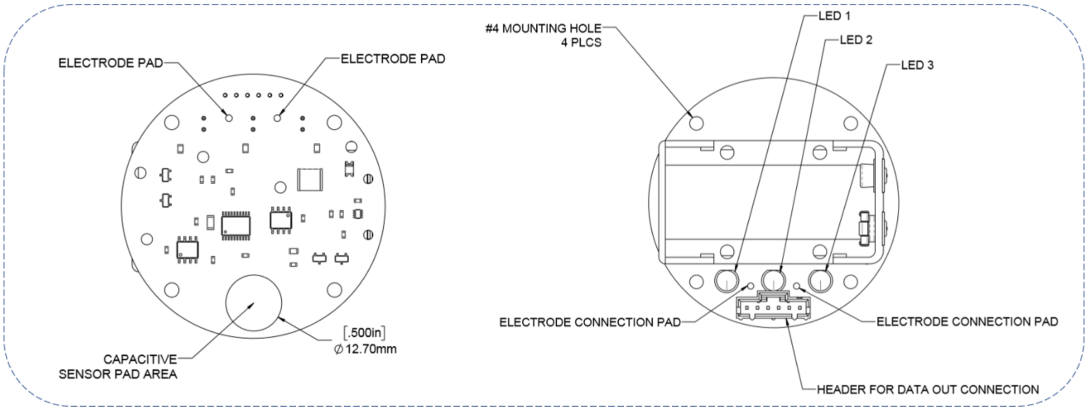


Figure 12: Main PCB Assembly

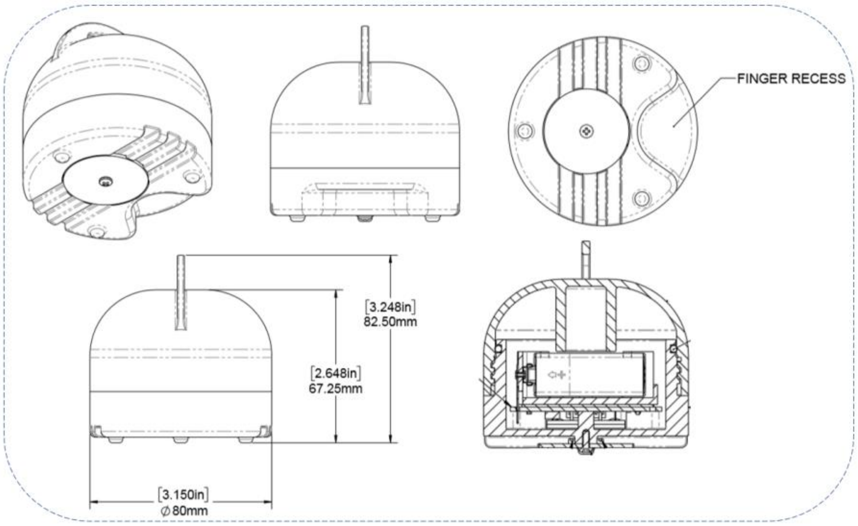


Figure 13: Device Enclosure Assembly

Project Time Line

Our project timeline changed pretty drastically throughout the semester as we began designing and testing. Throughout the testing process, we encountered numerous unforeseen problems that required us to continuously make design changes. From our initial testing of the header board, we realized that the switches we initially planned to use were not fast enough. Rather than create a new PCB board and have to wait for it to arrive, we rewired the header board and continued to test with it. Another major problem we encountered throughout the semester was the lack of access to a reliable silver testing machine. The machine we had planned on using did not run properly and continuously produced invaluable results. As a result, we could not perform all of the testing that we would have liked.

Overall, we did most of our design and testing in a parallel manner; however, for each part of the project, we executed tasks serially. While we worked on the hardware and software development in parallel, the individual development of the two occurred serially.

Kathryn primarily worked on creating the PCB layout and chose components for the device. She also worked with Mark to design the enclosure and ensure the components and PCB board would fit within the specifications. Isaac and Josh primarily wrote software to program the device and include all of the functionality set forth in the requirements presented to us by the Civil Engineering department. We all worked together to test and debug the device.

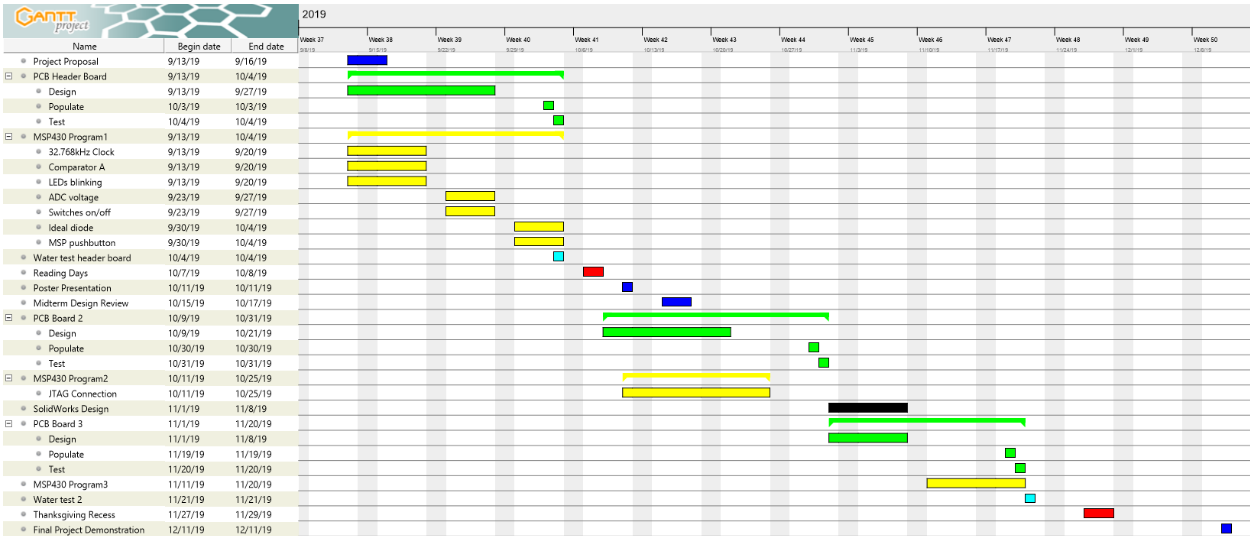


Figure 14: Gantt Chart Schedule (Proposal)

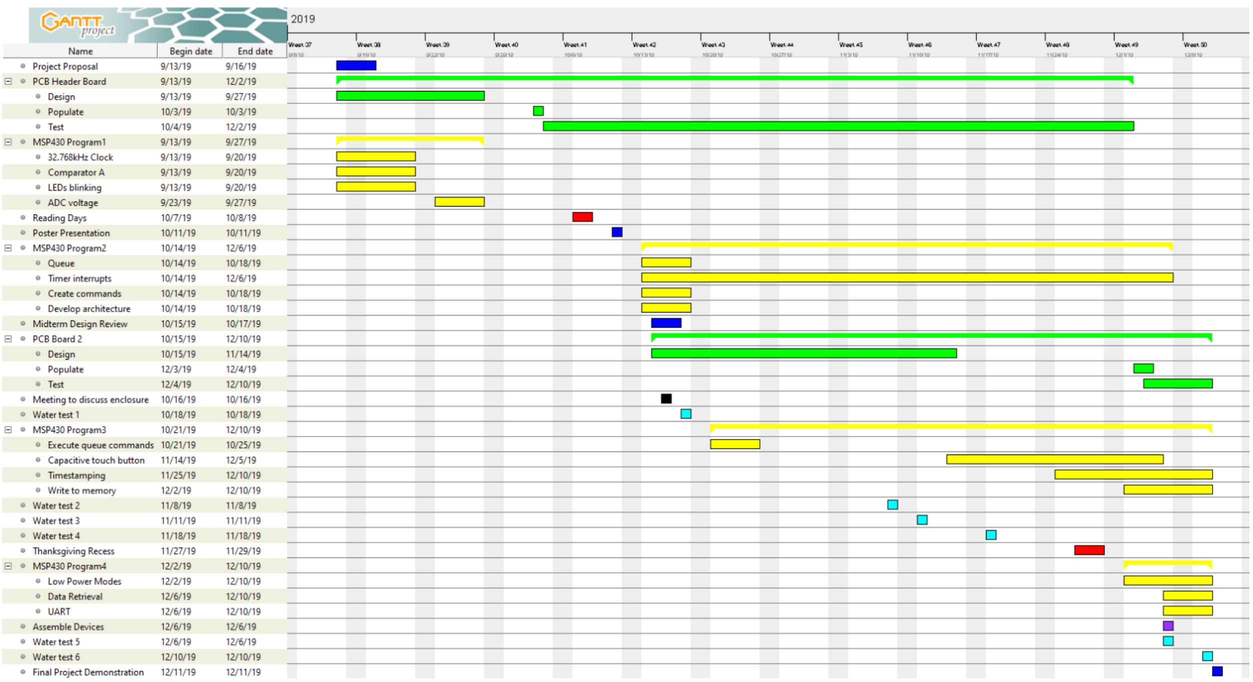


Figure 15: Gantt Chart Schedule (Final)

Test Plan

Our test plan changed significantly throughout the semester as problems arose. We began testing the basic functionality of our design using the header board that we created. The header board included the 9V battery connection, the LTC4412 ideal diode circuitry, the switches to charge and discharge the capacitor, the three LEDs and their NMOS controllers, and the comparator and ADC circuitry. During our initial testing, we found the switches to behave too slowly and opted to remove them for the final design, since we could achieve the same functionality using the LTC4412.

We first tested our design by implementing the software to charge and discharge the capacitor to the appropriate voltages. We charged the capacitor for a pre-set amount of time and used the comparator to control the switch from the discharge to charge state. In order to test that the ADC and comparator were working, the system was set to perform two charge/discharge cycles and store all ADC values in an array (measurements were taken at both the charged and discharged state). We set our Virtual Bench to make a single capture and then paused our test unit using the MSP430 debugger to view the values in the array. We then compared all the values for the max and min voltages on the capacitor measured by the ADC against the virtual bench. Additionally, we checked to make sure the capacitor started charging when the voltage read by the comparator around ½ VCC. It is important to note that the values read by the ADC and comparator on the MSP are stepped down by a voltage divider from the voltage on the capacitor because 9 volts would exceed the maximum threshold on MSP430 input pins. Finally, we checked to make sure that the calculated value for the total change in voltage for both cycles was correct.

After we tested the device and software with no load, we began testing using various resistors to simulate the load environment to ensure we received results that aligned with our calculations. We measured the change in voltage as well as the discharge time for a single cycle to determine the amount of charge that would be released and the total time it would take to release. All of the values aligned with our calculations and demonstrated that the same amount of charge would be released regardless of the load resistance.

Once we were satisfied with the results we received from the resistors as the load, we decided to test the device on various concentrations of synthetic ground water. However, due to the unreliability of the original silver testing machine, we did not receive accurate results for the first three tests. For the fourth test, we decided to switch machines. Although the machine provided much more accurate results, we received silver concentration values that ranged between 40-60 ppb. While the silver release was not constant across the different concentrations of synthetic ground water, the Civil Engineering department was satisfied with the results, since the silver levels were all still under 100 ppb. They believe the variability could easily be within the error of the analytical method.

When testing the final PCB design, we noticed that the LT3009 3.3V regulator was not outputting the 3.3V as expected. After re-reading through the datasheet, we discovered that we had not tied the shutdown pin high, so we had to add an additional wire to make the connection.

To test the capacitive touch sensor, we created an external RC circuit to test each individual part of the capacitive touch sensor system: GPIO, TimerA1 in capture mode, and the dynamic reference value. We first tested the interrupt on the GPIO by turning an LED on when the voltage of a GPIO pin was changed from LOW to HIGH and HIGH to LOW to verify both the charge and discharge stages of the capacitive touch sensor. Then for TimerA1, we tested to see if the TAR register value increased as expected. We had to use a very slow frequency for the clock because the TAR value would count up too fast for us to verify its functionality. Finally, to test the dynamic reference value, we had to implement a UART communication with a laptop so we could have a live feed of the reference value changing dynamically.

When we tested the assembled prototypes with synthetic ground water, we discovered that the cover over the electrodes impeded the silver release and prevented the device from performing properly. We tried to modify the cover by drilling holes into it, but the holes still did not allow enough water to come in contact with the silver electrodes for the device to behave ideally. As a result, we performed the final water test without the cover to ensure consistency across the three devices. All three devices released roughly the same amount of silver, which is what we had expected. We are currently working with Mark to make modifications to the electrode cover design to create a cover that will not affect the performance of the device or inhibit the silver release.

Final Results

Although we encountered countless problems throughout the development of the device, we were able to successfully produce a final product that releases an appropriate amount of silver into water. Although the silver concentration was measured to be higher than 50 ppb, the concentration is still under the EPA SMCL and WHO guideline of 100 ppb. Since we did not have access to a reliable silver testing machine for much of the semester, we were not able to perform the amount of water testing that we would have liked. We also need to research more about the chemical process of silver release in water to better understand the variability in the results that we obtained.

Sample Number	Unit Number	107 Ag Conc. [ppb]
1	1	71.78973
2	2	62.37285
3	3	70.46509

Table 1: Silver Results from Water Test

The three LEDs blink as we would expect. The yellow LED blinks while the device is releasing silver and remains blinking for 4 hours to indicate that the user must wait for the ions to have sufficient time to disperse in the water and sterilize it. The green LED blinks at the end of the 4 hour contact waiting period and remains blinking until the 12 hours have elapsed since the initial button press. The red LED flashes when the user tries to press the button before the 12 hour period has elapsed to indicate that it is too soon to begin the silver release process again.

The final capacitive touch sensor works as expected based on our success criteria. Once a human finger is in contact with the touch plate, it triggers a button press just as it would for a mechanical push button. However, the capacitive touch sensor does not function well when there is water in contact with the touch plate and finger at the same time. We expected this due to the nature of the touch sensor. Once the water is wiped off, however, the button works as expected.

The data exporting software also works as expected based on our success criteria. The Python software originally was expected to take care of the UART serial communication in addition to the conversion and export to a readable CSV file. However, we had problems with interfacing with the new UART to USB converter and changed our plan to take care of the UART serial communication with the flash memory using a 3rd party software, RealTerm, and make a Python program that has a GUI which would take the raw data file from RealTerm and converted it into a readable and understandable CSV file that the Civil Engineers can use to interpret the data at their field test ground.

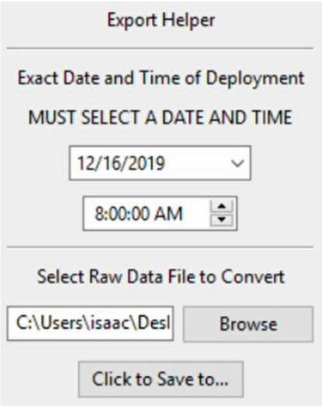
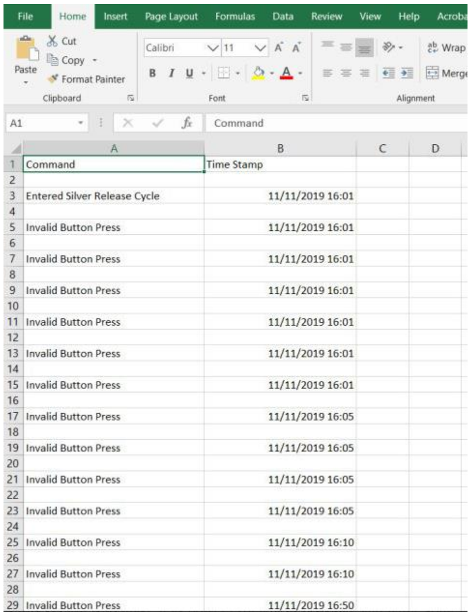


Figure 16: Data Export GUI



	A	B	C	D
1	Command	Time Stamp		
2				
3	Entered Silver Release Cycle	11/11/2019 16:01		
4				
5	Invalid Button Press	11/11/2019 16:01		
6				
7	Invalid Button Press	11/11/2019 16:01		
8				
9	Invalid Button Press	11/11/2019 16:01		
10				
11	Invalid Button Press	11/11/2019 16:01		
12				
13	Invalid Button Press	11/11/2019 16:01		
14				
15	Invalid Button Press	11/11/2019 16:01		
16				
17	Invalid Button Press	11/11/2019 16:05		
18				
19	Invalid Button Press	11/11/2019 16:05		
20				
21	Invalid Button Press	11/11/2019 16:05		
22				
23	Invalid Button Press	11/11/2019 16:05		
24				
25	Invalid Button Press	11/11/2019 16:10		
26				
27	Invalid Button Press	11/11/2019 16:10		
28				
29	Invalid Button Press	11/11/2019 16:50		

Figure 17: CSV File Output

One of the main goals for this system is to make it as power efficient as possible, and while some sacrifices had to be made on this front for purposes of both functionality and added feature requirements for the initial field-test of the device, we are continuing to make the device more efficient. Currently the system draws an average current of 1.2 mA, but we are continuing to evaluate the prototype design to improve power consumption. We are in the process of replacing the LMC6482 Op-Amp, which has a supply current around 1.2 mA; therefore, we expect our current consumption to be reduced to several microamps after we replace this part. We have ordered new op-amps from Digi-Key and plan to test the power consumption with them as soon as they arrive so that we can perform an accurate power analysis and determine the lifetime of operation of the device from a single 9V battery.

Costs

To produce 1 unit: roughly $26.50 for parts (excluding taxes and shipping) + $288.19 for enclosure + cost to assemble device

To produce 30 units (cost per device):
Roughly $100 per enclosure + $73.65 per device (includes parts and assembly) + $3.44 for silver wire = roughly $177.09 per device

To produce 10,000 units (cost per device):
roughly $12.38 for parts + cost of enclosure + cost to assemble device

The 30 units produced for the field test include components to store and retrieve data that would not be needed in the final manufactured product. Therefore, when we calculated estimates for parts to produce 10,000 units, we did not include those components. For the production of 30 units as well as the production of 10,000 units, we can utilize automated equipment such as a pick and place machine, which would decrease the cost for assembly.

Future Work

Future work on this project will need to take the results from the upcoming field tests into account in order to improve the user experience/operation of the device. Additionally, there are several changes that can be made to make the device less costly, more efficient, and more effective.

In order to make the device less costly the additional features that we added for the field test such as USB/UART communication and flash memory can be removed. Further, without the requirement for recording when the device is being used, it will be possible to decrease the device's power consumption by increasing the proportion of time that it spends in low power mode. Additionally, incorporating the release of coper ions into the water would increase the device's effectiveness in treating water. For those continuing work on this project we recommend thoroughly evaluating parts both on a cost and power consumption basis. Additionally, ensure you have a reliable and cost-effective way to measure silver concentration in any water you test the device on. Finally, we found using a FIFO queue to be the best approach for the embedded software design so we recommend continuing to use that structure.

References

[1] "Drinking Water: Equity, safety and sustainability, 2011," *UNICEF DATA*, 26-Jun-2018. [Online]. Available: https://data.unicef.org/resources/drinking-water-equity-safety-and-sustainability-2011-thematic-report/. [Accessed: 20-Oct-2019].

[2] "Grand Challenges - 14 Grand Challenges for Engineering." [Online]. Available: http://www.engineeringchallenges.org/challenges.aspx. [Accessed: 09-Dec-2019].

[3] "Silver as a drinking-water disinfectant." [Online]. Available: https://www.who.int/water_sanitation_health/publications/silver-02032018.pdf?ua=1. [Accessed: 20-Oct-2019].

[4] B. Ehdaie, C. T. Rento, V. Son, S. Turner, A. Samie, R. A. Dillingham, and J. A. Smith, “Evaluation of a Silver-Embedded Ceramic Tablet as a Primary and Secondary Point-of-Use Water Purification Technology in Limpopo Province, S. Africa,” *PLOS ONE*, 17-Jan-2017. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0169502. [Accessed: 20-Oct-2019].
[5] “The MadiDrop+,” *MadiDrop+*. [Online]. Available: https://www.madidrop.com/madidrop-1. [Accessed: 20-Oct-2019].
[6] “Secondary Drinking Water Standards: Guidance for Nuisance Chemicals,” *EPA*, 08-Mar-2017. [Online]. Available: https://www.epa.gov/dwstandardsregulations/secondary-drinking-water-standards-guidance-nuisance-chemicals. [Accessed: 20-Oct-2019].
[7] D. M. Threlkel, “NEMA Enclosure Types,” no. 17, p. 9.
[8] “NEMA/IEC Enclosure Ratings from Cole-Parmer.” [Online]. Available: https://www.coleparmer.com/tech-article/nema-iec-enclosure-ratings. [Accessed: 13-Sep-2019].
[9] Bureau of Indian Standards, *IS/IEC 60529: Degrees of protection provided by enclosures (IP CODE)*. 2001.
[10] “Silver,” *Water Quality Association*. [Online]. Available: https://www.wqa.org/learn-about-water/common-contaminants/silver. [Accessed: 09-Dec-2019].
[11] “Basic Information about Lead in Drinking Water,” *EPA*, 18-Nov-2019. [Online]. Available: https://www.epa.gov/ground-water-and-drinking-water/basic-information-about-lead-drinking-water. [Accessed: 09-Dec-2019].
[12] "International Standard | IEC 60950-1." [Online]. Available: https://webstore.iec.ch/preview/info_iec60950-1%7Bed2.0%7Den_d.pdf. [Accessed: 13-Sep-2019].
[13] *Acceptability of Electronic Assemblies*, IPC-A-610G, 2017.
[14] Method of and apparatus for disinfecting liquids by anodic oxidation with a silver anode, by V. Eibl and A. Reis. (1974, May 28). *Patent US3923632A.* Accessed on: Dec. 10, 2019. [Online]. Available: https://patentimages.storage.googleapis.com/f8/28/55/b911d7f11fe940/US3923632.pdf.
[15] Device for purifying water, by V. M. F. Dahlgren. (1981, Mar, 2). *Patent US4337136A.* Accessed on: Dec. 10, 2019. [Online]. Available: https://patentimages.storage.googleapis.com/c8/82/26/4731208a0297fc/US4337136.pdf.
[16] Leaching device for electrolyzed silver, by T. Okazaki. (1995, Dec. 6). *Patent US5614067A.* Accessed on: Dec. 10, 2019. [Online]. Available: https://patentimages.storage.googleapis.com/05/91/55/729bbafd81eb4b/US5614067.pdf
[17] “MSP430G2553 | MSP430G2x/i2x | MSP430 ultra-low-power MCUs | Description & parametrics.” [Online]. Available: http://www.ti.com/product/MSP430G2553. [Accessed: 09-Dec-2019].
[18] "LT3009 Series | 3μA I_Q, 20mA Low Dropout Linear Regulators." [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/3009fd.pdf. [Accessed: 09-Dec-2019].
[19] "LTC4412 | Low Loss PowerPath Controller in ThinSOT." [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/4412fb.pdf. [Accessed 09-Dec-2019].

[20] "ZVP3306F | SOT23 P-Channel Enhancement Mode Vertical DMOS FET." [Online]. Available: https://www.diodes.com/assets/Datasheets/ZVP3306F.pdf. [Accessed: 09-Dec-2019].
[21] "CB042D0474JBC | Film Chip Capacitor." [Online]. Available: http://datasheets.avx.com/cb-petht.pdf. [Accessed: 10-Dec-2019].
[22] "LMC6482 | CMOS Dual Rail-to-Rail Input and Output Operational Amplifier." [Online]. Available: http://www.ti.com/lit/ds/symlink/lmc6482.pdf. [Accessed: 09-Dec-2019].
[23] " SST25VF512A | 512 Kbit SPI Serial Flash." [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/25090A.pdf. [Accessed: 09-Dec-2019].
[24] "D1213A | 1 Channel Low Capacitance TVS Diode Array." [Online]. Available: https://www.diodes.com/assets/Datasheets/ds32185.pdf. [Accessed: 09-Dec-2019].
[25] " DMN3135LVT | 30V Dual N-Channel Enhancement Mode MOSFET." [Online]. Available: https://www.diodes.com/assets/Datasheets/DMN3135LVT.pdf. [Accessed: 09-Dec-2019].
[26] "SD1206S040S2R0 | Schottky Barrier Rectifier Diode." [Online]. Available: http://datasheets.avx.com/schottky.pdf. [Accessed: 09-Dec-2019].
[27] "uUSB-PA5-II | USB to Serial-TTL UART bridge converter." [Online]. Available: https://4dsystems.com.au/mwdownloads/download/link/id/313/. [Accessed: 09-Dec-2019].

Appendix

Index	Qty	Part Number	Manufacturer Part Number	Description	Unit Price	Ext. Price	Vendor
1	2	1276-1102-1-ND	CL10A105KA8NNNC	CAP CER 1UF 25V X5R 0603	0.1	0.20	DigiKey
2	3	1276-1033-1-ND	CL10B104JB8NNNC	CAP CER 0.1UF 50V X7R 0603	0.1	0.30	DigiKey
3	1	478-4537-1-ND	CB042D0474JBC	CAP FILM 0.47UF 5% 63VDC 2220	1.15	1.15	DigiKey
4	2	311-1059-1-ND	CC0603JRNPO9BN120	CAP CER 12PF 50V C0G/NPO 0603	0.1	0.20	DigiKey
5	1	1276-2082-1-ND	CL10B474KA8NFNC	CAP CER 0.47UF 25V X7R 0603	0.1	0.10	DigiKey
6	1	718-1118-1-ND	293D106X0010A2TE3	CAP TANT 10UF 20% 10V 1206	0.3	0.30	DigiKey
7	1	1276-1987-1-ND	CL10B222JB8NNNC	CAP CER 2200PF 50V X7R 0603	0.1	0.10	DigiKey
8	1	365-1181-ND	OVLFG3C7	LED GREEN CLEAR 5MM T/H	0.37	0.37	DigiKey
9	1	365-1183-ND	OVLFY3C7	LED YELLOW CLEAR 5MM T/H	0.2	0.20	DigiKey
10	1	365-1182-ND	OVLFR3C7	LED RED CLEAR 5MM T/H	0.2	0.20	DigiKey
11	2	D1213A-01SO-7DICT-ND	D1213A-01SO-7	TVS DIODE 3.3V 10V SOT23	0.36	0.72	DigiKey
12	1	478-7806-1-ND	SD1206S040S2R0	DIODE SCHOTTKY 40V 2A 1206	0.45	0.45	DigiKey
13	1	A33931-ND	5-103735-5	CONN HEADER VERT 6POS 2.54MM	1.72	1.72	DigiKey
14	2	DMN3135LVT-7DICT-ND	DMN3135LVT-7	MOSFET 2N-CH 30V 3.5A TSOT26	0.44	0.88	DigiKey
15	1	LTC4412ES6#TRMPBFCT-ND	LTC4412ES6#TRMPBF	IC OR CTRLR SRC SELECT TSOT23-6	3.44	3.44	DigiKey
16	2	ZVP3306FCT-ND	ZVP3306FTA	MOSFET P-CH 60V 0.09A SOT23-3	0.46	0.92	DigiKey
17	2	541-4021-1-ND	CRCW060347K0FKEAC	RES SMD 47K OHM 1% 1/10W 0603	0.1	0.20	DigiKey
18	2	311-390ARCT-ND	RC0805JR-07390RL	RES SMD 390 OHM 5% 1/8W 0805	0.1	0.20	DigiKey
19	1	311-470ARCT-ND	RC0805JR-07470RL	RES SMD 470 OHM 5% 1/8W 0805	0.1	0.10	DigiKey
20	1	P402KDACT-ND	ERA-6AEB4023V	RES 402K OHM 0.1% 1/8W 0805	0.36	0.36	DigiKey
21	1	P124KDBCT-ND	ERA-3AEB1243V	RES SMD 124K OHM 0.1% 1/10W 0603	0.35	0.35	DigiKey
22	1	311-5.1MGRCT-ND	RC0603JR-075M1L	RES SMD 5.1M OHM 5% 1/10W 0603	0.1	0.10	DigiKey
23	1	LT3009ESC8-3.3#TRMPBFCT-ND	LT3009ESC8-3.3#TRMPBF	IC REG LINEAR 3.3V 20MA SC70-8	2.42	2.42	DigiKey
24	1	296-28430-1-ND	MSP430G2553IPW20R	IC MCU 16BIT 16KB FLASH 20TSSOP	2.41	2.41	DigiKey
25	1	LMC6482AIMX/NOPBCT-ND	LMC6482AIMX/NOPB	IC OPAMP GP 2 CIRCUIT 8SOIC	1.89	1.89	DigiKey
26	1	SST25VF512A-33-4C-SAE-ND	SST25VF512A-33-4C-SAE	IC FLASH 512K SPI 33MHZ 8SOIC	0.44	0.44	DigiKey
27	1	XC1617CT-ND	ECS-.327-12.5-34B-TR	CRYSTAL 32.7680KHZ 12.5PF SMD	0.54	0.54	DigiKey
28	1	59K0295	1294	BATTERY HOLDER 9V PC PIN	1.71	1.71	Newark
29	1	75935A13		VHB Tape, 1" Wide, 15 Feet Long, Black		0.34	McMaster-Carr
30	1	9452K153		O-Ring 3/32 Fractional Width, Dash Number 146		0.21	McMaster-Carr
31	2	9452K111		O-Ring 1/32 Fractional Width, Dash Number 001		0.08	McMaster-Carr
32	4	91735A102		Screws 4-40 Thread, 1/4" Long		0.30	McMaster-Carr
33	3	91802A105		Oval Head Screws 4-40 Thread, 3/16" Long		0.16	McMaster-Carr
34	2	41456G9	AA41456G9	Silver Wire, 1.0mm dia, 99.9% (Metal basis), 2in	1.72	3.44	FisherScientific
					Price for unit	26.50	

Table 2: 1 Unit Parts Cost

Index	Qty	Part Number	Manufacturer Part Number	Description	Unit Price	Extended Price USD
1	1	1613-1118-ND	UUSB-PA5-II	MICROUSB PROGRAMMING ADAPTER	20.4	20.40
2	1	S1112EC-05-ND	PREC005SBAN-M71RC	CONN HEADER R/A 5POS 2.54MM	0.19	0.19
3	1	A9CAG-0604F-ND	A9CAG-0604F	FLEX CABLE - AFG06G/AF06/AFE06T	2.64	2.64
4	1	S1012E-03-ND	PEC03SAAN	CONN HEADER VERT 3POS 2.54MM	0.22	0.22
5	1	311-330HRCT-ND	RC0603FR-07330RL	RES SMD 330 OHM 1% 1/10W 0603	0.1	0.10
6	1	A33161-ND	5103308-2	CONN HEADER VERT 14POS 2.54MM	1.36	1.36
					Price per unit	24.91

Table 3: Adapter Board Parts Estimate

Qty	Unit price	Extended Price USD
30	*73.65	2209.5

Table 4: 30 Units Cost Estimate

*Price does not include enclosures and silver wire

Index	Qty	Part Number	Manufacturer Part Number	Description	Unit Price	Ext. Price	Vendor
1	10000	365-1181-ND	OVLFG3C7	LED GREEN CLEAR 5MM T/H	0.29	2,900.02	DigiKey
2	10000	365-1183-ND	OVLFY3C7	LED YELLOW CLEAR 5MM T/H	0.16	1,600.00	DigiKey
3	10000	365-1182-ND	OVLFR3C7	LED RED CLEAR 5MM T/H	0.16	1,600.00	DigiKey
4	20000	311-1059-2-ND	CC0603JRNPO9BN120	CAP CER 12PF 50V C0G/NPO 0603	0.00932	186.40	DigiKey
5	20000	1276-1102-2-ND	CL10A105KA8NNNC	CAP CER 1UF 25V X5R 0603	0.00656	131.20	DigiKey
6	28000	1276-1033-2-ND	CL10B104JB8NNNC	CAP CER 0.1UF 50V X7R 0603	0.00836	234.08	DigiKey
7	2000	1276-1033-1-ND	CL10B104JB8NNNC	CAP CER 0.1UF 50V X7R 0603	0.0121	24.20	DigiKey
8	10000	718-1118-2-ND	293D106X0010A2TE3	CAP TANT 10UF 20% 10V 1206	0.0644	644.00	DigiKey
9	8000	1276-1987-2-ND	CL10B222JB8NNNC	CAP CER 2200PF 50V X7R 0603	0.00672	53.76	DigiKey
10	2000	1276-1987-1-ND	CL10B222JB8NNNC	CAP CER 2200PF 50V X7R 0603	0.0088	17.60	DigiKey
11	10000	LTC4412ES6#TRMPBFTR-ND	LTC4412ES6#TRMPBF	IC OR CTRLR SRC SELECT TSOT23-6	1.834	18,340.00	DigiKey
12	18000	ZVP3306FTR-ND	ZVP3306FTA	MOSFET P-CH 60V 0.09A SOT23-3	0.1313	2,363.40	DigiKey
13	2000	ZVP3306FCT-ND	ZVP3306FTA	MOSFET P-CH 60V 0.09A SOT23-3	0.1716	343.20	DigiKey
14	20000	541-4021-2-ND	CRCW060347K0FKEAC	RES 47K OHM 1% 1/10W 0603	0.00353	70.62	DigiKey
15	10000	P402KDATR-ND	ERA-6AEB4023V	RES 402K OHM 0.1% 1/8W 0805	0.03499	349.86	DigiKey
16	10000	P124KDBTR-ND	ERA-3AEB1243V	RES SMD 124K OHM 0.1% 1/10W 0603	0.03469	346.92	DigiKey
17	10000	311-5.1MGRTR-ND	RC0603JR-075M1L	RES SMD 5.1M OHM 5% 1/10W 0603	0.00196	19.64	DigiKey
18	10000	LT3009ESC8-3.3#TRMPBFTR-ND	LT3009ESC8-3.3#TRMPBF	IC REG LINEAR 3.3V 20MA SC70-8	1.33	13,300.00	DigiKey
19	10000	296-28430-2-ND	MSP430G2553IPW20R	IC MCU 16BIT 16KB FLASH 20TSSOP	0.99	9,900.00	DigiKey
20	10000	LMC6482AIMX/NOPBTR-ND	LMC6482AIMX/NOPB	IC OPAMP GP 2 CIRCUIT 8SOIC	0.8025	8,025.00	DigiKey
21	9000	XC1617TR-ND	ECS-.327-12.5-34B-TR	CRYSTAL 32.7680KHZ 12.5PF SMD	0.23635	2,127.15	DigiKey
22	1000	XC1617CT-ND	ECS-.327-12.5-34B-TR	CRYSTAL 32.7680KHZ 12.5PF SMD	0.28688	286.88	DigiKey
23	20000	311-390ARTR-ND	RC0805JR-07390RL	RES SMD 390 OHM 5% 1/8W 0805	0.00356	71.28	DigiKey
24	10000	311-470ARTR-ND	RC0805JR-07470RL	RES SMD 470 OHM 5% 1/8W 0805	0.00356	35.64	DigiKey
25	8000	1276-2082-2-ND	CL10B474KA8NFNC	CAP CER 0.47UF 25V X7R 0603	0.01449	115.92	DigiKey
26	2000	1276-2082-1-ND	CL10B474KA8NFNC	CAP CER 0.47UF 25V X7R 0603	0.02005	40.10	DigiKey
27	18000	DMN3135LVT-7DITR-ND	DMN3135LVT-7	MOSFET 2N-CH 30V 3.5A TSOT26	0.12802	2,304.43	DigiKey
28	2000	DMN3135LVT-7DICT-ND	DMN3135LVT-7	MOSFET 2N-CH 30V 3.5A TSOT26	0.16731	334.62	DigiKey
29	9200	478-4537-2-ND	CB042D0474JBC	CAP FILM 0.47UF 5% 63VDC 2220	0.3564	3,278.88	DigiKey
30	800	478-4537-1-ND	CB042D0474JBC	CAP FILM 0.47UF 5% 63VDC 2220	0.51612	412.90	DigiKey
31	9000	478-7806-2-ND	SD1206S040S2R0	DIODE SCHOTTKY 40V 2A 1206	0.1125	1,012.50	DigiKey
32	1000	478-7806-1-ND	SD1206S040S2R0	DIODE SCHOTTKY 40V 2A 1206	0.13133	131.33	DigiKey
33	10000	59K0295	1294	BATTERY HOLDER 9V PC PIN	0.86	8,603.44	Newark
34	100	75935A13	Each	VHB Tape, 1" Wide, 15 Feet Long, Black	33.94	3,394.00	McMaster-Carr
35	200	9452K153	Packs of 50 each	O-Ring 3/32 Fractional Width, Dash Number 146	10.48	2,096.00	McMaster-Carr
36	200	9452K111	Packs of 100 each	O-Ring 1/32 Fractional Width, Dash Number 001	3.66	732.00	McMaster-Carr
37	800	91735A102	Packs of 50 each	Screws 4-40 Thread, 1/4" Long	3.73	2,984.00	McMaster-Carr
38	300	91802A105	Packs of 100 each	Oval Head Screws 4-40 Thread, 3/16" Long	5.21	1,563.00	McMaster-Carr
39	40	41456G9	AA41456G9	Silver Wire, 1.0mm dia, 99.9% (Metal basis), 25m	846.00	33,840.00	FisherScientific
						123,813.97	
					Price per unit	12.3814	

Table 5: 10,000 Units Parts Estimate