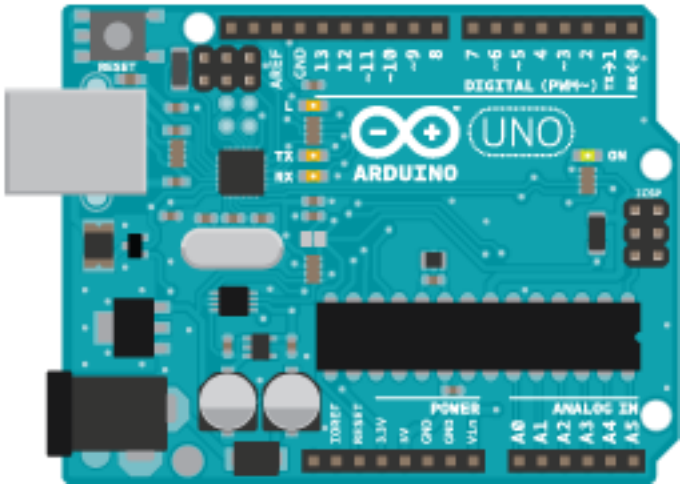
Computer Engineering Senior Project

Cal Poly San Luis Obispo

Isopropyl Alcohol Pump for Printed Circuit Boards using an Arduino



Advisor:

Andrew Danowitz

By:

Lawrence Zhu

December 2016

Table of Contents

1. Introduction

1.1. Project Overview

For most smart-phones users, water damage is one of the most common source of damage they have to deal with. Dropping these phones in any type of liquid can lead to water damage or potentially destroy it from the inside out. The general solution found online to repair water damage is sticking the phone into a bag of uncooked rice overnight, hoping that it can turn on again the next day. However, this process can result with a device still containing small droplets of water after being removed from the bag. The combination of water and copper within the pcb cause oxidation to occur. With the PCB cleaner, there is a higher chance that the phone will survive. The PCB cleaner uses a pumping system with isopropyl alcohol flowing into the device, flushing out trapped water. Isopropyl alcohol can help resolve many issues after the device is exposed to liquid.[1] It is non conductive as well so none of the electronics can get damaged. This cleaner uses 99% isopropyl alcohol, with the other 1% being water. Even though a majority of the liquid being used is alcohol, I would recommend the user to dry off the pcb as much as possible before and remove any type of batteries that can cause a flow of electricity. This accounts for any additional liquid on the board and makes sure the board doesn't have electricity running through it.

Currently the only similar technology is ultrasonic PCB cleaners which are quite expensive. These devices uses high frequency waves to creates cavitation, which is the formation of bubbles that implode and "blast away" contaminants on the surface of the board.[11] The problem with these cleaners is that device such as have a price point of hundreds of dollars. My device, the PCB cleaner, is more of a viable option for the average consumer.

1.2. Stakeholders

The target market for this product is composed of consumers who wish to be able to clean smartphones without having to pay a large sum of money in doing so. A majority of our population has a type of cell phone or tablet, we have a broad range of customers. From techies that have an interest in cleaning electronics to corporate employees wishing to retrieve files from a wet phone, my PCB cleaner a simple product that doesn't dent the wallet.

This device would also affect the companies that create ultrasonic cleaners. If there were a different solution to market for a fraction of the price, there would be more competition for their products. Since there are only a few companies in the market that specialize in PCB cleaning devices, these companies can manipulate the price freely.

In addition to smartphone users, people that wish to disinfect tools or objects could benefit from this product as well. Isopropyl alcohol, or rubbing alcohol as it is more commonly known as, is a general disinfectant that kills off bacteria, funguses, and viruses.

1.3. Project Goals and Deliverables

The project outcome is to build a device that is capable of containing roughly any size tablet or phone and repair it from water damage. The deliverables for this project are the prototype device and a report containing all the information about the device.

2. Background

Printed circuit boards are defined as the board base for physically supporting and wiring of surface-mounted and socketed components.[3] These boards are printed and etched in with thin, conductive pathways that serve as the link to components such as integrated circuits, resistors, and transistors. Before the invention of the PCB, components had to use a point to point wire connection. With the PCB, it creates a cleaner workspace without the need for excess wire clutter. Most electronics today use at least one PCB in its circuitry. Some key and essential features of PCBs include giving mechanical support to electronic components, serve as conductors of heat transfer, and provide additional space for components.[4]

While technology in our electronics advances, the size of the PCB began to decrease significantly. As top of the line electronics today are packed with more new features, the space required to fit in on a PCB becomes smaller and smaller. There is now a challenge to squeeze the most processing capacity, functionality, and memory into this small circuit board. These small size boards are more likely to experience failures. Some problems that can occur are fragile pieces and electric currents that potentially jump from one pathway to another if etched close enough to each other. PCBs are required to have a minimal amount of distance between these etching so current does not jump over accidently. However, if substances like tap water were to come in contact with the board, then ions in the water give a pathway for currents. Water also has an aftereffect where the PCB might corrode onto itself.

Corrosion is permanent issue that affects PCB efficiency if not treated properly. When copper traces, which carry the current of all electrical components on the PCB, are corroded, the performance and lifespan of the pcb start to degrade. This is because the electrolytes in the liquid causes new traces to exists, wears down the original trace, and forms a green build up. Think of a two roads where drivers go in opposite directions and a new road were to be place in between these road. These new traces can rewire the board by shorting it out or even redirecting signals to wrong places on the pcb. There are two main types of corrosion that can cause a device to fail: Galvanic and Dendritic Corrosion [5]; both occurs when new traces are being formed when it is

not suppose to. Dendritic corrosion is the worse of the two types since it has the ability to redirect current.

Though isopropyl alcohol cannot fix corroded boards affected by water damage, it can prevent the problem from occurring in the first place. Corrosion only occurs when the water damage sets into the phone. If a person were to take quick action with their phones and try to get rid of the water as soon as possible, there is a greater chance of the device surviving.

3. Engineering Specifics

3.1. Previous Versions

Over the course of this senior project there have been multiple prototypes of this design, with their own strengths and weaknesses. The previous types of versions include:

3.1.1. Silica Gel

Silica gel is most commonly found in small packets placed in containers and dry foods. These packets contain small beads that are placed in a sealed, porous bag. Surrounding water vapor gets trapped within these tiny holes, keeping the material in the package dry and at a low humidity. Since silica is safe enough to be placed with food, this material is more than safe to keep with electronics without worry.

With this implementation of the project, a box is filled with silica packets and the phone is placed in the center, completely submerged in the beads. An electronic is then used to lock the box from being opened and to secure the device in the box. The box is locked to prevent users from trying to turn on their piece of electronics before the drying process is complete.

Some limitations of using these silica packets are that the beads can contain only so much moisture before being saturated. To fix this you would have to bake the beads in a 300℉ oven until the moisture is gone. Also the beads are only capable of bringing down the humidity of the surrounding area down to about at most 40%. An issue that arose is the time required for the silica to have an effect on the phone or device. The recommended amount of time to place a phone in a bag of dry rice is 3-5 days. During this time the liquid is just sitting on the components, causing corrosion to occur. A device that has to stay in this box filled with silica gel for as long as a day might as well be considered broken. Even though silica gel has a better absorption rate, the time required causes other problems that can't be fixed without rewire or replacing parts on the board.

3.1.2. Heater

The most simple way to get rid of the water is by drying it off, which was the idea for this implementation. Using a heat lamp, the water on the board evaporates from the board. This

implementation also calls for the user to take apart the device, separating it into pieces so everything can be dried out thoroughly. Since the temperature required to evaporate water is lower than the melting point of the pcb board, including all the solder joints and components, the board can withstand the heat without degrading the performance or melt. This process is similar to placing the an electronic device in an oven on low to keep the pcb at a constant heat.

The process starts with the phone, which is placed into the box. Once inside, the heat lamp will turn on. Once the evaporation is done, the phones stays in the box and a fan turns on to cool down so the user can pick it up. Once cool, the dryer box notifies the user that the process is done and the phone is ready to pick up.

The problem with this implementation is that if the liquid isn't water, soda for instance, residue might get left behind and might damage the board as well. This also can cause more problems since more dust can get stick and cause future problems like overheating. Tearing down electronics and separating it into pieces is another problem for the average consumer. A device, such as an iphone, contain very small screws and thin wires that can be broken easily if not taken apart carefully. There is no point in saving a phone when the user can potentially break it more trying to fix it.

4. Final Design Process

4.1. Schematics / Design pictures

***Figure 1**: Complete setup of PCB cleaner*

Figure 2: *Top view of black box*

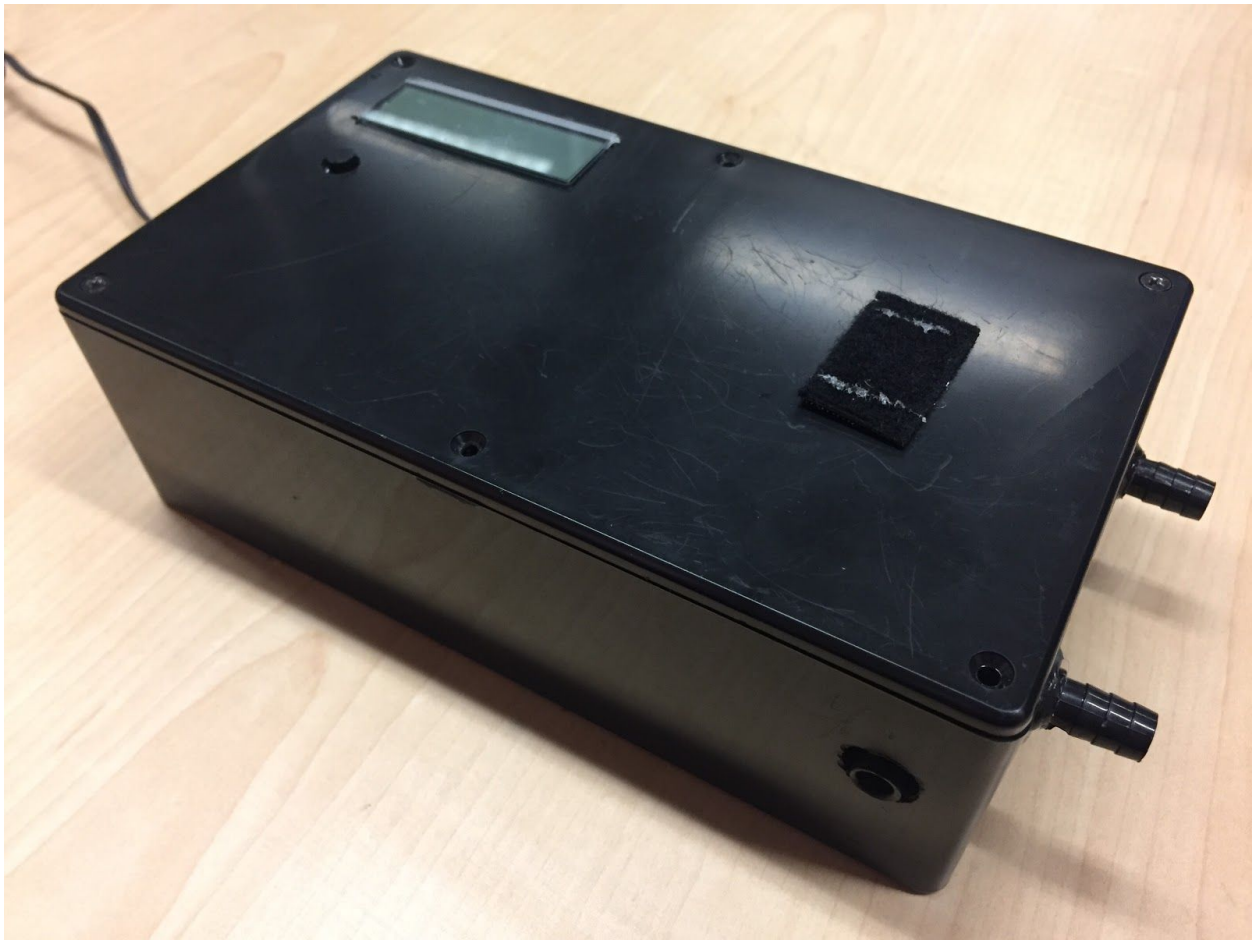


Figure 3: Side View of Black Box

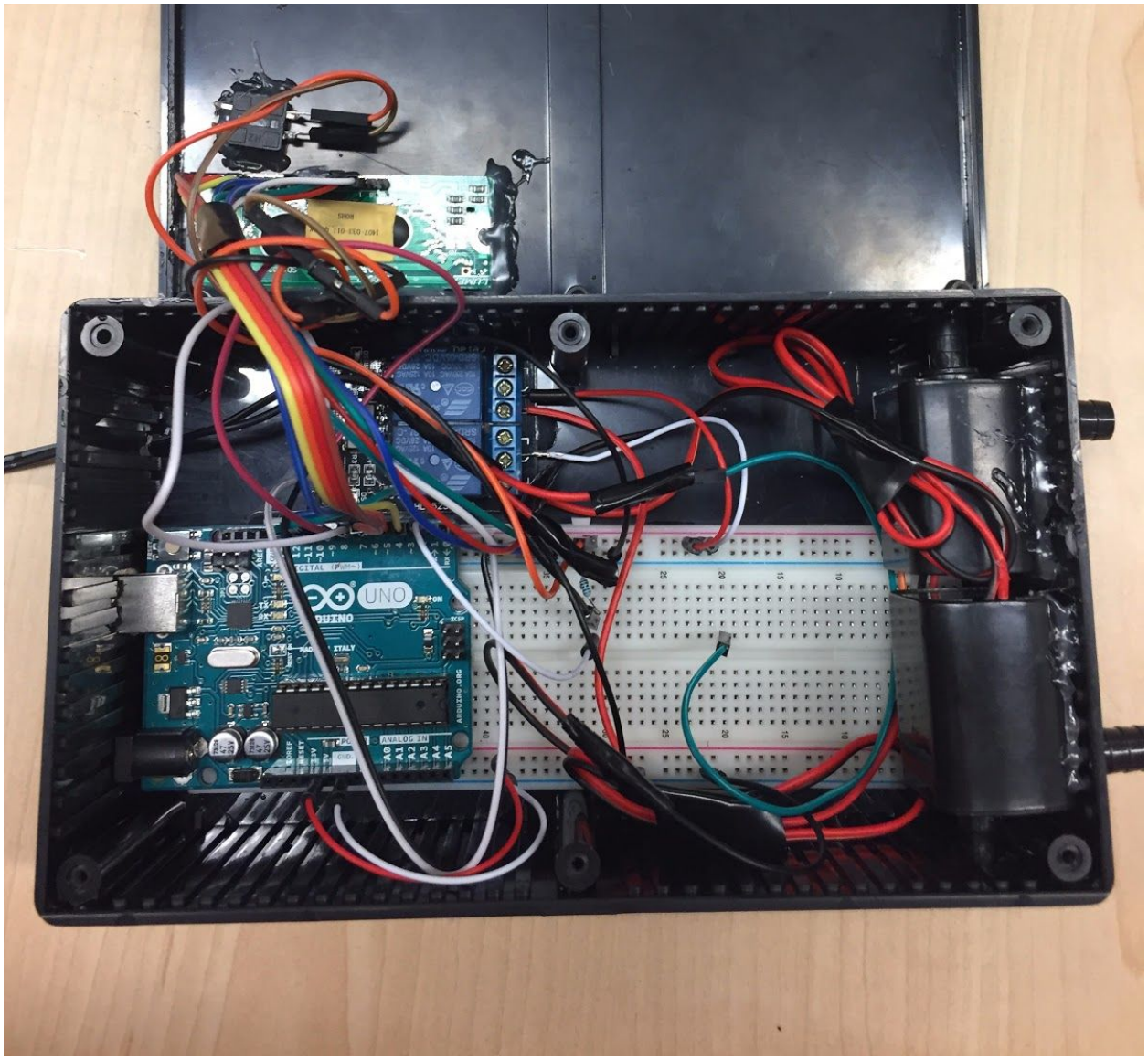


Figure 4: *Inside Black Box*

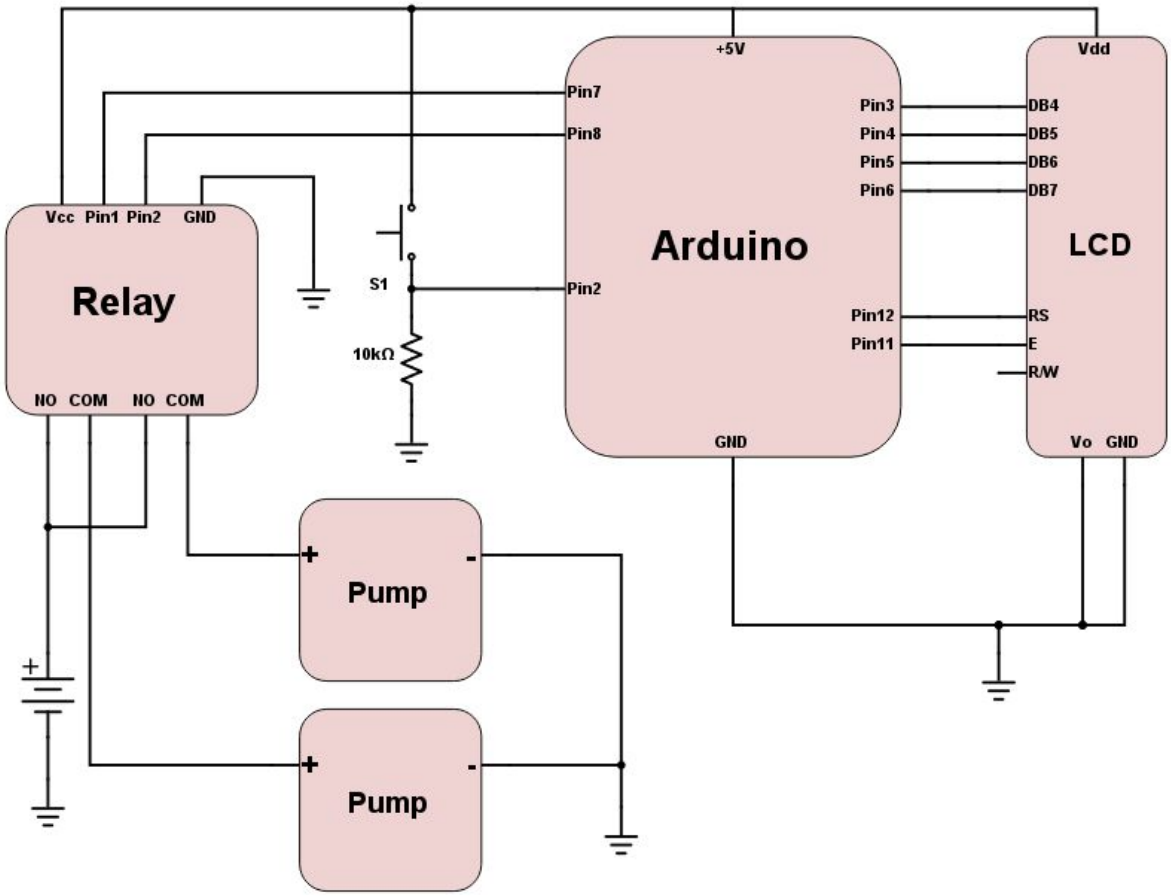


Figure 5: *Schematic Design*

4.2. Setting up Arduino

For this project, I will be using an Arduino as the main communicator of the components and users. The Arduino has a versatile library and large community forum filled with many step by step guides and libraries for general use. This device works across all popular platforms such as Windows, Mac, and Linux. For its retail price of $24.95, it is small enough to be hidden in any space and more than powerful enough to run the whole system through a usb connection.

4.2.1. Libraries

This project requires the arduino the have additional libraries to function. The software to code arduino can be found in the references. [6]

When the application is installed, the Timer1 library needs to added in. Navigate to: Sketch -> Include Library -> Manage Libraries

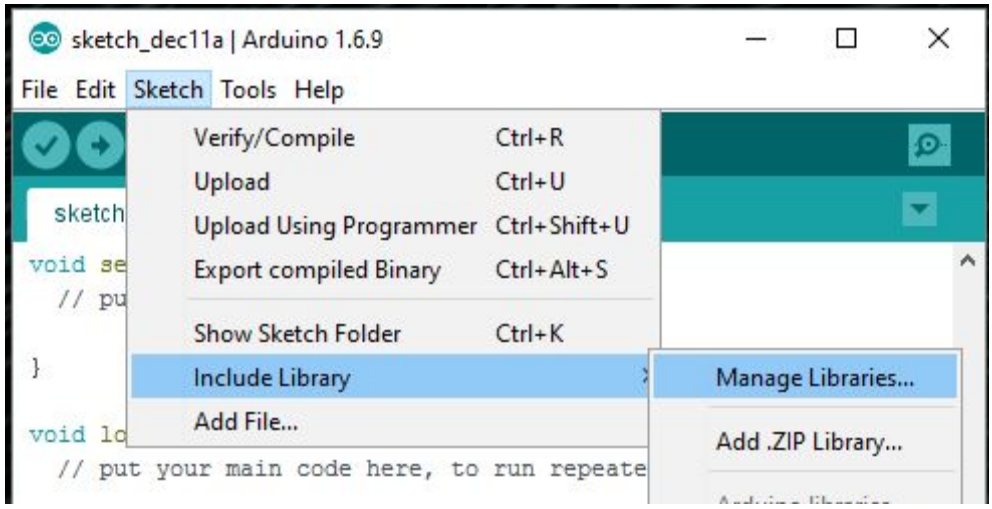


Figure 6: *How to find Libraries*

Once the window pops up you can find and install the Timer1 library. [9]

4.2.2. Debouncer

When pushing a button on the arduino, the microcontroller actually see more than one press. This leads to problems since the button changes the state of the lcd screen but if it sees more than one press, it just jumps pass all the states. To fix this a debouncer is added.[10] The debouncer checks to see if the button is pressed. Once the button is pressed a timer starts and this accounts for a long and short button press. The duration of the timer is longer than a button press so this just filters out extra presses.

6.3. Software Flow Chart / State Diagram

Figure 7: *State diagram of program*

The figure above shows the general flow of the program. The program starts at the welcome screen state, where the only thing a person can do to get to the next state is to press the button. This is where the user is prompted to place the phone is in the alcohol bath prior to pressing the button. Once the button is pressed the state jumped into the main screen state. During this state both pumps turn on and the phone is then submerged in a bath that has flowing alcohol from a reservoir. From here the main screen can go through two action but end up at the end screen. This is due to the fact that the timer is set the five minutes. Once the timer has gone down to zero the state automatically goes into the end screen state. This allows the user to not worry about how much time is needed and can do other tasks. Also if the user wished to end the program early they can press the button to overwritten the program to go straight to the end screen.

5. Overall System Analysis

With the PCB Cleaner, the user is able to wash their phone in an alcohol bath and clean it from debris such as dirt and dust. Since isopropyl alcohol doesn't evaporate quickly unless spread to thinly or heated, it could potentially last more than a few rounds.

Some problems that occurred while taking on this project were the pumps. I was not sure if the pumps I had were strong enough or the values connected to the pump were air tight but the pumps were able to pull up the water. I've tried to increase the voltage from 5V to 12V but it still hasn't shown signs of improvement in water control. A quick solution would be to purchase a different set of pumps. Another goal of this cleaner was to contain any device such as a 18" tablet. This goal was not met since the box that contains the pcb is too small but this can be changed since the vinyl pumps can be removed and switched around. This potentially allows for the user to switch between a large and small chassis for their pcb boards.

6. Future Considerations / Implementations

6.1. More Powerful Water Pumps/Filtering system

For this project the pumping mechanism could definitely be improved on. I believe that the pumps that I have used are not as efficient as others. Another thing that can be added is a filtering system so no piece can be lost within the pumps or reservoir. If fragments were to get lodged into the pumps also this could lead to damage to the system. Something like a cage could work since it can contain all the electronics in one area while making it easier to retrieve later on from the alcohol bath.

6.2. Speaker vibrating at 25 kHz

Comparing my PCB cleaner to commercial one, theirs uses ultrasonic waves that vibrate the alcohol to move water molecules. Though my implementation is done through the flow of a pump, there isn't nearly as much movement compared to ultrasonic waves. A future implementation would probably be a hooked up speaker of some short that can emit high frequencies that can simulate what the PCB cleaner simulates.

7. Bill of materials

***Table 1**: Bill of Materials*

Item Description	Quantity	Price($)	Link
Push button	1	$2.95	Link
5V relay	1	$5.98	Link
DC Water Pumps	2	$19.98	Link
16x2 LCD screen	1	$6.56	Link
Project Box	1	$7.81	Link
99% Isopropyl Alcohol	1	$6.51	Link
Vinyl Tubing	1	$5.30	Link
Clear Storage Tote	1	$4.49	Link
Total	$59.58		

Even though the price of the materials was higher than expected, there are many ways to get around this situation. The LCD screen are real necessities since the arduino software provides a serial monitor to print out values.

8. References

1. “Using alcohol to fix or revive a phone dropped in water.” http://www.smartmobilephonesolutions.com/content/using-alcohol-phone-dropped-in-wate

2. Uses of Isopropyl Alcohol: http://www.livestrong.com/article/191247-common-uses-for-rubbing-alcohol/

3. Defintion of pcb: http://whatis.techtarget.com/definition/printed-circuit-board-PCB

4. Importances of printed circuit boards: http://hardhero.com/the-importance-of-printed-circuit-boards/

5. “What really kills electronics, water or corrosion?” http://www.hzo.com/what-really-kills-electronics-water-or-corrosion/

6. Drying with Silica Gel: http://www.romwell.com/books/craft/SilicaGel.htm

7. Arduino software for computer: https://www.arduino.cc/en/Main/Software

8. Liquid Crystal Display Library: https://www.arduino.cc/en/Reference/LiquidCrystal

9. TimerOne Library: http://playground.arduino.cc/Code/Timer1

10. Debouncing Tutorial: https://www.arduino.cc/en/Tutorial/Debounce

11. How to Clean PCBs: https://www.sfcircuits.com/pcb-school/clean-pcbs-ultrasonic-cleaning

9. Appendix

9.1. Arduino Code: main.ino

```
/*
 * Author: Lawrence Zhu
 * Title:  CPE Senior Project
 * File:   main.ino
 */
#include "senior_project.h"

// (RS,E,D4,D5,D6,D7,D8)
// connects to the pins above on the LCD
LiquidCrystal lcd(12, 11, 3, 4, 5, 6);

void setup()
{

  // Sets up LCD screen
  lcd.begin( 16, 2 );
  lcd.clear();

  // Set up relay
  pinMode( PUMP_ONE_PIN, OUTPUT );
  pinMode( PUMP_TWO_PIN, OUTPUT );

  // Set up button
  pinMode( BUTTON_PIN, INPUT );

  // Sets up Serial for debug
  Serial.begin( 9600 );

  //Sets up Interrupt Timer every second
  Timer1.initialize( 1000000 );
  Timer1.attachInterrupt( timerIsr );

  // Initially turn pumps off
  turnPumpOff( PUMP_ONE_PIN );
  turnPumpOff( PUMP_TWO_PIN );

}

void loop()
{

  int reading = digitalRead( BUTTON_PIN );

  // Checks to see if the is a button push
  if ( reading != lastButtonState )
    lastDebounceTime = millis();

  // Checks if the button push was just a long button push to account
  // for noise and human error
  if (( millis() - lastDebounceTime ) > debounceDelay )
  {
    if ( reading != buttonState )
```

```
    {
      buttonState = reading;
      if ( buttonState == HIGH )
      {
        curState = (curState + 1) % STATES;
      }
    }
  }
  lastButtonState = reading;
}

// prints timer onto the lcd screen
void printTimer()
{
  // Prints the countdown timer
  lcd.print( ( int ) minutes );
  lcd.print( ":" );
  // checks to see if the value is a single digit otherwise prints it
  if ( seconds < 10 )
  {
    lcd.print( "0" );
    lcd.print( ( int ) seconds );
  }
  else
  {
    lcd.print( ( int ) seconds );
  }
}

// This function shifts over a string to the left by 1
// and plance the first char of the string to the end
void change( char * msg )
{
  char buff[ strlen( msg ) ];

  memcpy( buff, &msg[1], strlen( msg ) - 1 );
  buff[strlen( msg ) - 1] = msg[0];
  memcpy( msg, buff, strlen( msg ) );
}

void printScreen()
{
  switch( curState )
  {
    case 0:
      // First Screen: Welcome screen
      lcd.print( welcome_message );
      change( welcome_message );
      break;

    case 1:
      // Second Screen: Starts circulation of pumping.
      lcd.print( second_message );
      change( second_message );

      turnPumpOn( PUMP_ONE_PIN );
      turnPumpOn( PUMP_TWO_PIN );

      // print how time is left
      lcd.setCursor( 0, 1 );
```

```
      lcd.print( "Timer[" );
      printTimer();
      lcd.print( "]" );
      break;

    case 2:
      // Final Screen: Turns off pumps and displays end message
      turnPumpOff( PUMP_ONE_PIN );
      turnPumpOff( PUMP_TWO_PIN );

      lcd.print( end_message );
  }
}

// Updates the timer by incrementing down a second
void updateClock()
{
  if ( curState == 1 )
  {
    if ( seconds == 0 )
    {
      if ( minutes == 0 )
      {
        curState = 2;
      }
      if ( minutes > 0 )
      {
        minutes--;
        seconds = 59;
      }
    }
    else
    {
      seconds--;
    }
  }
}

// Interrupt timer that happens every second
// clears the screeen and updates the clock
void timerIsr()
{
  lcd.clear();
  lcd.setCursor( 0, 0 );
  printScreen();
  updateClock();
}
```

9.2. Arduino Code: senior_project.h

```
/*
 * Author: Lawrence Zhu
 * Title:  CPE Senior Project
 * File:   senior_project.h
 */
#include <LiquidCrystal.h>
#include <TimerOne.h>
#include <string.h>

// Pins used on the Arduino board for button and pumps
#define BUTTON_PIN    2
#define PUMP_ONE_PIN  7
#define PUMP_TWO_PIN  8

// different states used for LCD
#define STATES 3

// define functions to turn pumps on and off
#define turnPumpOff( x ) digitalWrite( x, HIGH );
#define turnPumpOn( x )  digitalWrite( x, LOW );

// global variables for debouncer
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;

// Used for the button and debouncer
volatile int buttonState = 0;
volatile int curState = 0;
int lastButtonState = LOW;

// Timer set to run for 5:00 minutes
int minutes = 5;
int seconds = 0;

// welcome message
char welcome_message[] =  " Welcome Screen. Press button to start.";
char second_message[] =   "Starting circulation of pumping.";
char end_message[] =      "Done. Please remove device";
```