Wax Bee Gone:

A Safe, Effective Solution To Ear Wax Removal

By: Robert Shenon and Shanvir Dhinsa

EE 460 - Senior Project Preparation

Electrical Engineering Department

California Polytechnic State University

San Luis Obispo

Spring 2017

Table of Contents

List of Tables

List of Figures

Acknowledgements

We would like to thank Tina Smilkstein for being a supportive and helpful Senior Project Advisor and giving us the idea for the senior project in the first place. We'd also like to acknowledge the help we received from the EE Technical Support Group in providing us with the 3D Printer for our final prototype and the Dremel. Without the EE Technical Support Group, this project would have been much harder than it needed to be. Plus, IEEE's supplies and concepts as well as Innovation Sandbox's assistance for the first prototype helped us out immensely.

Abstract

This project sought to combine conventional methods such as scraping and water pressure to removal ear wax, but with electronic automation to removal the potential for human error as much as possible. Through the use of a 9V battery, and Arduino microcontroller, DC motors, and other circuitry, this project has developed a prototype for a device that can properly remove ear wax without any discomfort or risk to the user. Much of this project involved learning how to properly manipulate and provide the necessary power requirements DC motors from an Arduino and a 9V battery. Additionally, creating a 3D printed shell to house all the components in a watertight case was one of the more difficult parts of the project. This device can be used by any age group or level of experience. From our empirical data this product satisfies the need that was not fulfilled when the project began.

1) Background and Introduction

There are currently 18 million people in the United States that suffer from excess ear wax [4]. The idea behind this project is to create a truly effective and non-intrusive ear wax removal device. There are many ear wax removal devices on the market, but each has their own flaws including risk of infection, discomfort, or even damage to the ear canal. Current devices do not properly implement modern technology, so this device will include different methods of cleaning like motorized scraping and water spray technology that is computer controlled. Overall, this will provide as an excellent solution to ear wax removal which currently does not exist

Before one can examine the importance of a product like this, it will require one to understand what ear wax is and whether it helps or hinders one's health. The production of ear wax is a natural part of human health and is created by the glands in the ear in order to protect the ear canal from small particles, such as dust. Ear wax production is different from person to person and mostly depends on genetics. For most people, it is not a problem, but for many others, excessive ear wax buildup can cause hearing loss and bacterial infection which can lead to dryness, itchiness, or cause severe health risks. However, ear wax is still an important part of one's health, so when it comes to ear wax removal it is important to make sure there is still a reasonable amount of ear wax left. [1]

There are many methods currently available to remove ear wax. One of the more popular methods is "cotton tipped applicators" such as Q-tips. However, these devices are not recommended by any means. Even though they can remove ear wax, they can also push ear wax deeper into the ear canal and cause additional hearing loss or possibly even long term damage. If pushed too deep into the ear canal, they can even cause "perforation," which is when the breaking of the ear canal. (see Figure 1). Online, one can find many ear wax removal devices that are designed to allow the user to scrape away the ear wax in a safer manner by having soft, more flexible tips. While superior to Q-tips for the purpose of ear wax removal, they can still push ear wax deeper into the ear canal if used improperly and are therefore not recommended by doctors. One of the more effective solutions out there include various liquid solutions to rinse out the ear canal and soften or loosen the ear wax. Some of these solutions include olive oil, baby oil,

hydrogen peroxide, and even just distilled water. This method works quite well, but by completely rinsing out the ear, one can actually remove too much ear wax and leave their ear exposed to the small particles such as dust and bacteria that ear wax is created to protect against. Additionally, if the user has wet ear wax, just using a liquid rinse might not remove enough of the ear wax. The only surefire way to remove ear wax both effectively and safely is to see a medical professional and let them use a variety of scraping and rinsing methods to eliminate excessive ear wax. The obvious problem with this is the time and costs inconveniences associated with seeing a medical professional, especially if the ear wax buildup if frequent. [1]

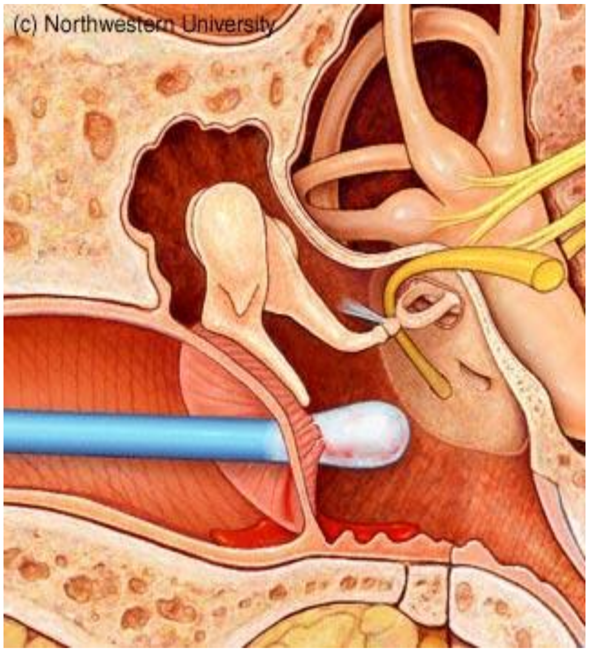

Figure 1.1: Example of why Cotton Tipped Applicators are unsafe for ear wax removal

Therefore, the idea behind this project is to find something that accomplishes the effectiveness of a medical professional in removing ear wax but without the hassle. To accomplish this, we will test various untried methods that utilize modern technology to find the best way to allow individuals to remove ear wax safely. We believe that the current ear wax removal devices could benefit from some of the recent advances in biomedical devices. While we have not reached the testing and prototyping phase of this project, there are various possible improvements including:

- Advanced sensors to allow for greater visibility
- Heating lamps to weaken ear wax to allow for easier removal,
- Automated scraper for ear wax removal without risk of human error
- Implementing solutions to remove ear wax and allow for solution retrieval
- Small vacuum to remove debris from ear canal

Again, we think an ideal device would not just have one method, such as scraping or a liquid solution, but would have multiple stages to optimize the process.

1.1 What is wrong with Present Solutions?

Present solutions can worsen an individual's case opposed to fixing it. Whether its Q-tips or a hydrogen peroxide rinse, these solutions have negative health effects like damaging or drying up the patient's ear canal. Furthermore, going to a medical professional can be a costly operation that many people can't afford to get done regularly.

1.2: The Proposed Solution:

The answer to many of the issues mentioned is simply to a rinse and scraper ear wax solution into one. Both methods work partially and have their own flaws, but by combining the best of both ideas, one can achieve an optimal solution. The current solutions outside of seeing a medical professional, run the risk of harming an individual's ear canal in some way shape or form. Our device would be the safest and best solution second to seeing a medical professional.

1.3 Background Information on Direct Motor (DC) Motors:

One of the more important parts of the project was getting the proper DC motors for our purposes. Unfortunately, due to our lack of knowledge on DC motors, we order the wrong ones and made our job much harder than it needed to be.

DC Motors are usually characterized from the maximum voltage you can put into it for an extended period of time. For example, a 3V DC Motor can maintain a voltage less than 3 for an extended period of time while a 7V motor can maintain a voltage less than 7 for an extended period of time. However, unlike silicon chips, they can go above the given voltage threshold. DC motors are very robust and can receive a voltage higher than their given limit without suffering

damage if not for extended periods of time. The last thing to keep in mind with DC motors is that their Rotations Per Minute (RPM) is adjustable depending on the voltage the motor receives.

For our project, we started with a 3V DC motor that we assumed would have low RPM, which was all we needed for our specifications. However, a 7V DC motor would have been able to have a low RPM as well, and would have been easier to adjust the speed. Also, we found that a 7V DC motor actually have a lower current draw as a 3V DC motor for the same RPM. This is important for the scope of our project where we had limited current draw from our components.

If we made a more researched selection on the type of DC motor for our power supply, we would have had easier success for our project.

1.4 Background Information on Pulse Width Modulation (PWM):

In addition to adjusting the voltage of the motor directly from the source, a much easier way to adjust the RPM of a DC motor is to use PWM. This is the concept of adjusting the duty cycle of a rectangular waveform. Normal DC current has a duty cycle of 100%. However, if you adjusted the duty cycle to 50%, the waveform would be only half the time, and off half the time, thereby adjusting (or modulating) the pulse. For a DC Motor, the voltage it sees is directly related to the duty cycle using the equation:

Voltage Seen by DC Motor = DC voltage * duty cycle.

For our purposes, we could use PWM to adjust the DC voltage and with a 5V signal we could have the motor “see” and voltage we need it to. Therefore, we could adjust the RPM of the motor to fit our specifications.

2) Design

2.1 Block Diagram Level 0

The main design from this project had to do with deciding what we needed to use to safely remove earwax. From the research above, we came to the conclusion that a scraper and a water pump would achieve this goal. With simple controls and rinse solution for the water pump, we would have all the inputs required for our system. This is shown visually in Figure 2.1 and explained in detail in Table 2.1

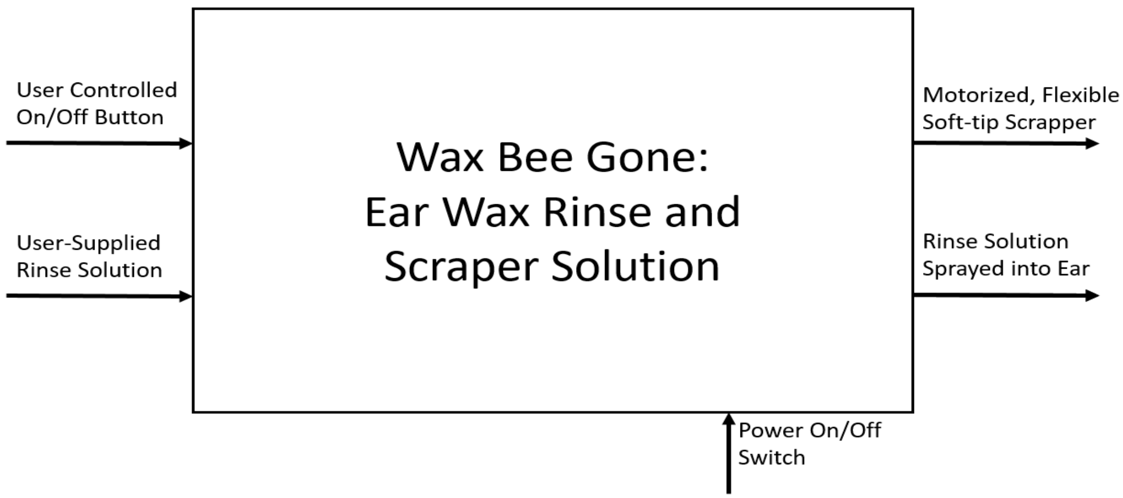

Figure 2.1: Block Diagram Level 0

Module	Ear Wax Rinse and Scraper Solution
Inputs	- User Controlled On/Off Button - User Supplied Rinse Solution - Power On/Off Switch
Outputs	- Motorized, Flexible Soft-tip Scraper - Rinse Solution Sprayed into Ear
Functionality	The product will clean one’s ear canal of excess ear wax with a simple button press safety and effectively.

Table 2.1: Ear Wax Rinse and Scraper Solution Specification

2.2 Block Diagram Level 1

The system itself requires the ability to take the simple inputs and process what is needed to remove the ear wax. To achieve this, we determined that a microcontroller would control the motors for the automated scrapper and water pump. All three of these would be powered by a 9V power supply that could be controlled by the user for complete on/off of the system. This is shown visually in Figure 2.2. Table 2.2 explains the use of the microcontroller in directing the functions of the motors from the inputs. Table 2.3 explains the use of the water pump water's use in outputting the necessary liquid to clean the ear. Table 2.4 elaborates on the use of the DC motor to provide an automated scrapper tip from the device. Table 2.5 elaborates on the power supply in powering all components within the product when turned on by the user.

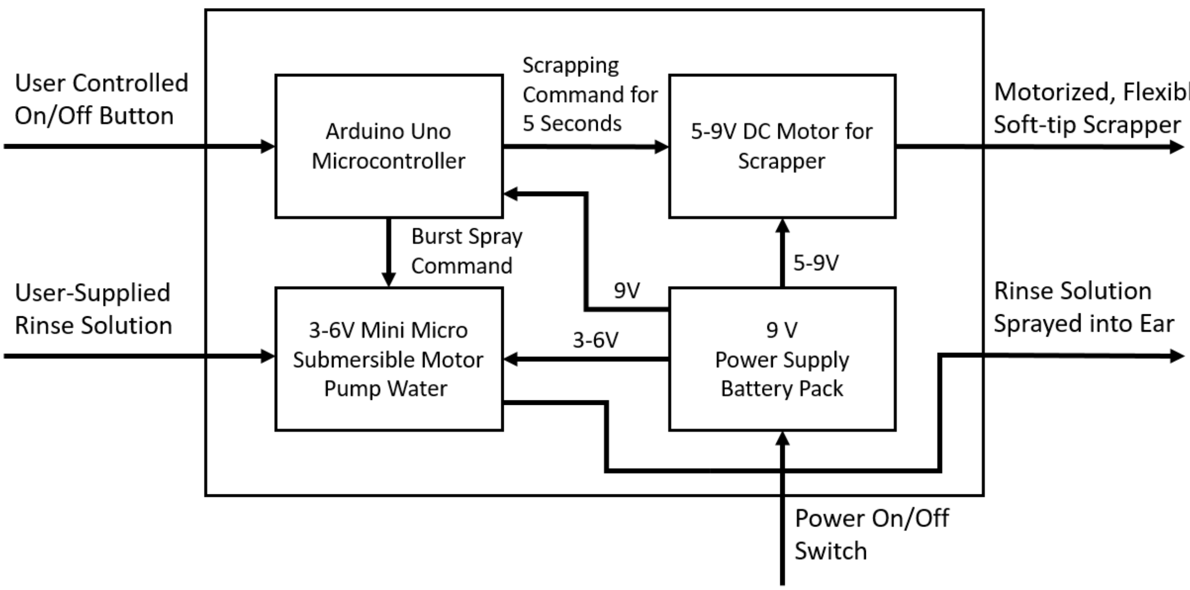

Figure 2.2: Block Diagram Level 1

Module	Arduino Uno Microcontroller
Inputs	- User Controlled On/Off Button - 9V Power
Outputs	- Scraping Command for 5 Seconds - Burst Spray Command
Functionality	This will be the controller of our product. The Arduino Uno simply needs to output the (binary) on and off controls to provide proper timings for the water spray and the scraper.

Table 2.2: Arduino Uno Microcontroller Specification

Module	3-6V Mini Micro Submersible Motor Pump Water
Inputs	- Burst Spray Command - 3-6V (depending on water flow)
Outputs	- Rinse Solution Sprayed into Ear
Functionality	This will spray the rinse solution safely into one's ear in order to soften the ear wax for removal. The liquid will be in short bursts in order to allow for draining between to avoid clogging one's ear.

Table 2.3: 3-6V Mini Micro Submersible Motor Pump Water Specification

Module	5-9V DC Motor for Scraper
Inputs	- Scraping Command for 5 Seconds. - 5-9V (depending on desired RPM)
Outputs	- Movement of Motorized, Flexible Soft-Tip Scraper
Functionality	After the solution rinse in the ear canal, the scraper will properly scrape away the softened ear wax without removing too much. This will have enough strength to get rid of the ear wax without damaging the ear canal

Table 2.4: 5-9V DC Motor for Scraper

Module	9V Power Supply
Inputs	- Power On/Off Switch - 9 V Battery supplied beforehand)
Outputs	- 9V for Arduino Uno - 3-6V for Motor Pump Water - 5-9V for Motor for Scraper
Functionality	Using a battery, this will power all the subsystems of the product for a minimum of 10 hours of "on" use.

Table 2.5: 9V Power Supply Specification

2.3 Shell Design

A major component to this project involved designing and constructing our 3D shell. Our shell needed to be big enough to house an Arduino, 9 volt battery back, DC motor, Water pump, and h bridge shield that communicates with the Arduino. Furthermore, all of the components needed to fit in a secure, safe, and appropriate manner. The initial design that we produced can be observed in the image below.

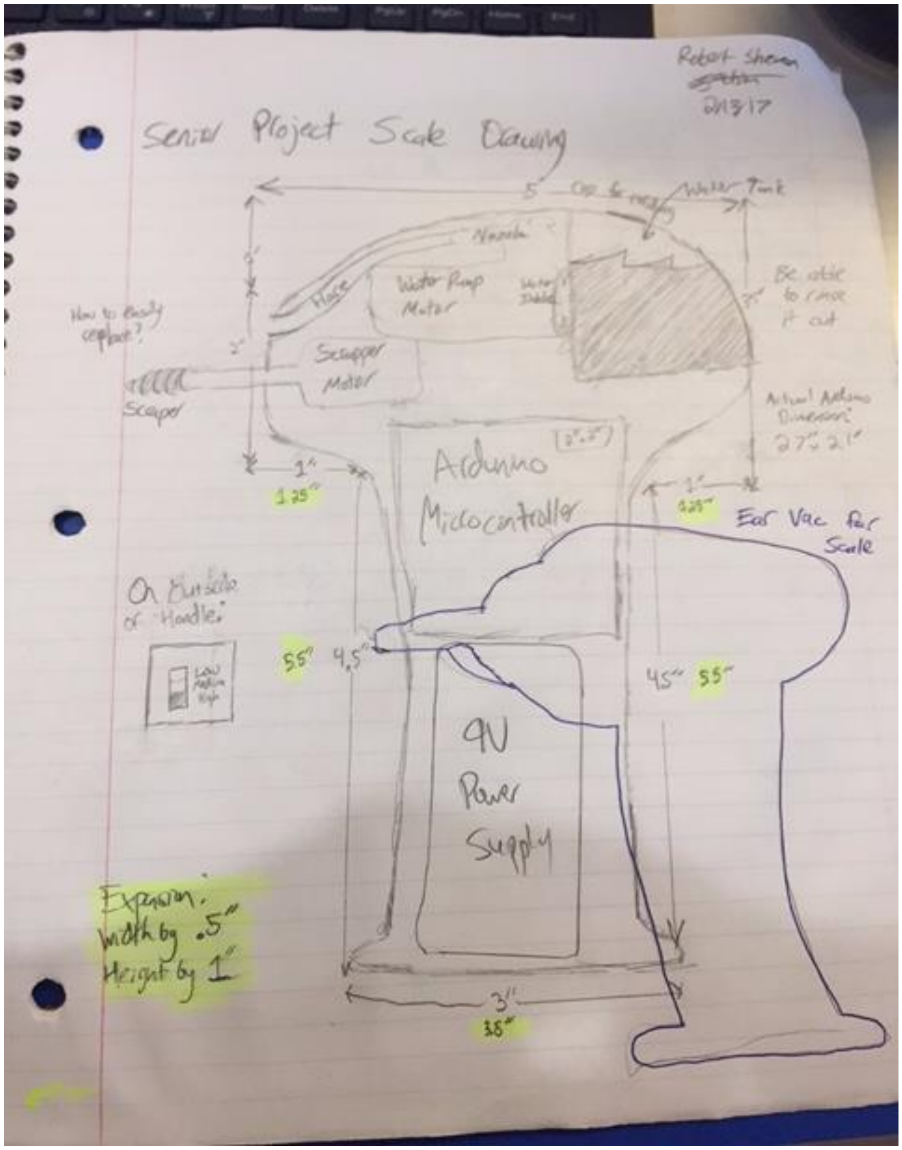

Figure 2.3: First Initial Design Sketch

As you can observe from the image above, we planned to have the DC motor towards the front of the gun with the tip of the scraper sticking out of the barrel. The motor pump would then be placed at the top of the shell, covering the hole that has water coming from the water container. These two components would be controlled by the Arduino micro-controller that is powered by a 9V power supply. Two switches would be used to allow the motor devices to be user controlled. Each switch having two settings, low and high.

Figure 2.4: 3D Model of Shell

The adjusted design now has the water motor placed directly within the storage container. The switch design was replaced with a button design for more user controllability. An L298N H-bridge is added to drive both the DC motor and water pump. The only downside to this design is that it is not very wieldy or ergonomic. Due to time constraints, we were forced to demo with our prototype shell. If allotted more time, a more physically comfortable and aesthetic shell could have been designed.

2.4 Water Pump and Water Storage

In order to implement a liquid solution for our project, we needed to conduct a variety of research in regards to pumping water into the ear canal, ideal voltage ranges for the pump, and pump design. To begin, we needed to research the human response to having water in the ear. As talked about in our research from EE 460, we found that pumping too much water into an individual's ear can cause swimmer's ear, a short term infection of the ear that causes inflammation and reduction of hearing. To prevent this symptom, we knew that we had to be very careful and not flood the ear with too much water.

In design our shell, we initially planned to have the water storage area connected to the back of the water pump through a watertight seal. This is to allow for a removal water storage area, as mentioned above that could be reconnected to the main body. However, we quickly realized that if the water pump was separate from the water storage, we would have problems with creating a watertight seal. This had to do with the fact that we could not close the case and then seal it from the inside. The required changes to the water pump and water storage is discussed in our Testing and Results section.

2.5 Switches and Buttons

Our initial black box design for this product had our buttons and switches communicating directly to the Arduino. This would allow for the user to set the required power setting (low, medium, high) voltage settings to the motor. However, following the redesign of our scraper and pump motors, we found having buttons that communicated to the Arduino and then the motors to be more difficult and somewhat unnecessary. The L298N H-bridge had 4 inputs and 2 enable inputs. The enables were fixed for motor A and B, and the first two inputs were used for picking the direction of motor A while the second two inputs were for the direction of motor B. For our purpose the directions did not matter much, but without extra inputs, having buttons feed into the Arduino before going to the motors wasn't as easy. This use of buttons in the design would change as our constraints developed in this project.

2.6 DC Motor

From our research from EE460, we knew that for the purpose of ear wax removal we did not need a strong motor. Therefore, we looked for a cheap, small, low voltage motor for our purposes. This lead us to finding a 3V motor that could have its RPM easily adjusted with voltage.

From early testing we also found that there was not just one right RPM for cleaning the ear. Because of this we decided it would be better to create 3 settings rather than just one: low, medium, and high. Low is .6V which can still scrape, but does not hurt the outside of the ear. Medium is still .8V and does the job, but can be sensitive to the outside of the ear. High is 1.1V which can scrape the ear wax out of the inside of the ear well, isn't too loud, but will slightly hurt outside of the ear.

Further on in the project, we found that three settings was harder to implement than initially planned as well as less necessary than our previous testing pointed to. Additionally, we found that the 3V motor was actually not the best choice of motor for our purposes, which is discussed more in depth in the Background and the Testing and Results sections.

3) Testing and Results

3.1 DC Motor

One of the more complicated parts of this project turned out to be implementing the simple .5V to 3V DC Motor. From our research, we knew that we wanted a small, low power motor. We wanted a small size so that we could easily fit it in our shell and a low voltage to so it be more easily powered. However, we came to learn that a smaller motor with a lower voltage range does not mean that it would be easier to control or have lower power usage. These aspects came to light later on in our experimentation process and report.

To start, it should be noted that we, as a senior project group, did not have much experience with either the Arduino or DC motors. While we learned much about how the Arduino powers motors and how DC motors themselves works, much of this was not known when we were working on the project. This is heavily discussed in our background portion.

During our initial planning in EE460, we had the idea to have the DC motors powered by the 9V battery. Upon determining the voltages, we could simply step down the voltage with fixed dc voltage regulators ordered online. However, since both our motors were planning on being less than 5V, we quickly discovered that we could get 0-5V DC from the Arduino with the correct code, which would work perfectly for our needs.

After additional research, it soon because obvious that a microprocessor could not directly power a motor, so we moved onto using an L293D motor driver. This motor driver was a simple integrated circuit that could provide around 1A of output current and give us 4.5V to 36V. While this did not get the exact output voltage we needed for this motor, the output current was a vast improvement from what we current we got from the microprocessor itself. While we had some successes on getting the output voltages desired from the H-bridge, when we connected the H-bridge circuitry to the motor, we only got a quick succession of buzzing sounds. After testing the motor, we found that for 1.1 V from the motor, as mentioned in initial test plans required a 1.5A output which was too much for our H-bridge to handle. When we tested the H-bridge circuitry connected to the motor, we found it would climb to 1A and then give out before it could get the necessary power to the motor.

Since our L298 didn’t work out, we wanted to try out some other circuitry for the Arduino to power motors online. Because of the Arduino’s popularity with hobbyists online, there were many circuits that are well explain with given diagrams. One of these circuits we tried to replicate was using a Power MOSFET. This properly powered the motor with the 5V from the Arduino (using USB at the time), but gave us a voltage of only around 1V and steadily dropped. Also, the article we used did not properly get into the theory behind how or why the circuit works as it does and how to adjust the circuitry to meet our specs. Therefore, we came to realize that the H-bridge circuitry would not be able to satisfy our requirements. While this idea did give us a running motor from our Arduino, which was more progress than we had initially, we knew this was not what we needed to reach our specifications.

Following more research and advice from fellow students, the next logical step was to use the Arduino L298N Dual H-Bridge Motor Controller Module. The advantage this H-bridge is connected directly to the 9V battery as well as the Arduino, which we were not previously able to do. More importantly, we could bet 3A for two motors, which near perfectly fit our specifications. The main problem with this approach was that the L298 only gave us a voltage range from 5V-12V, which needed to be lower for our 5V motor. While, we knew that PWM should technically be able to turn a 5V signal lower for our 3V motor, in practice it wasn’t work so we conceded that idea.

To get from 5V to 1.5V, we chose to use a LM317 Linear Voltage Regulator, which could turn out 5V input into any voltage between 1.25V and 37V. It even had an output power of at least 1.5A, which was more than enough for our circuitry. Plus, the LM317 circuitry for our purposes was given to us from the datasheet and only required two resistors and a simplistic equation (see Figure 3.1).

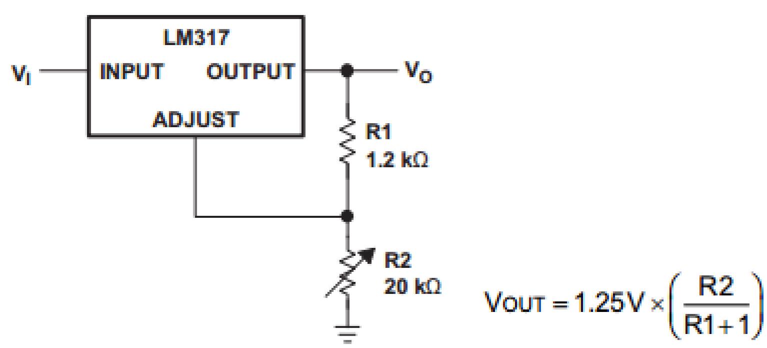

Figure 3.1: LM317 Datasheet Layout and Equation

Our initial testing using a multimeter and various resistors gave us voltages from 1.25-3V, which was ideal for this motor. As almost expected at this point, when we connected the motor to the output, we were unable get the current draw we needed to properly power the motor. Unfortunately, it turned out that we ordered LM317LZs. These linear voltage regulators operated very similar to LM317, but only had a current output of around 150mA. From testing the circuit with a motor and a power supply, we were able to get up to 185 mA which was still nowhere close to what we needed for our specific motor.

When talking to some mechanical engineers, they explained to us that using PWM, we should be able to turn our 5V into whatever voltage we needed for our motors. Unbeknownst to us until then, unlike silicon ICs that could burn out if too high of a voltage was put into them, DC motors only burned out if they were given too much voltage for too long. While we did agree with the theory behind it, we were still unable to get PWM to work with our small 3V motor. However, when testing with a 7V DC motor, we found that PWM allowed us to get practically any speed we needed.

At first we found this very confusing because we thought that the 7V motor, which was larger in size, we need more power. After more testing, we found that our initial assumption could not have been more wrong. Instead the 7V motor was only using our .6A of current while our 3V motor was using 2.5A of current for just a 2.5V input. This was the a key turning point in our project. Small voltage motors used a heavy current load which was much, much larger than for high voltage motors.

We found the problem with using PWM with our 3V motor was that even though the DC motor only saw the PWM voltage, that being the input voltage times the duty cycle, for the input voltage of 5V, the motor required a current greater than 3A which the H-bridge could not provide. Therefore, the PWM worked for the 7V motor that did not exceed the current load, but the small motor we planned on using required an excessive amount of current.

Even though we had a working motor, it no longer fit our shell. It was very close, but not close enough and we did not want to print an additional shell just to fit the new motor. Unfortunately, with the discovery that our 3V motor would no longer do the job with the equipment we had, we decided to fit the motor into our shell by using the Dremel on the inside edge and simply make room.

Now, we had a motor that could spin the scraper tip at any speed we needed, be powered by our H-bridge and Arduino, and perfectly fit inside the shell. The main takeaway from the scraper motor portion of the design was we should have put more research into DC motors, how they work, and finding the perfect DC motor for our specifications. We initially took the wrong approach and just went with a motor that looked like it would work for our device. This mostly had to do with our limited knowledge and us not knowing what we did not know, but again, this could have been remedied with more research put in.

3.2 Water Pump

Through much testing and trial, we concluded that the ideal voltage range for the motor was around 3V. This voltage range provided adequate pressure and water volume, ensuring that the ear would be damp enough for the scraper but not completely immersed in water. This would feed through a tube that would be fed through a nozzle and into the ear. Early testing with a power supply demonstrated that if the water pump motor was submerged in water, would would be able to eject water through the tube at any speed we might need.

As mentioned in the Design section, the main problem had to do with sealing the water storage area to the back of the water pump. While we tried multiple solutions to fit out original design, we found that the best solution was to actually put the water pump in the water storage area itself and just feed out the hose and the wires to power the motor. Even though we had to expand the hole from the water pump to fit the hose and the wires, we could much more easily seal the water in the water storage section and then send the hose directly through the shell to the front and the wires to the H-bridge/power supply.

Additionally, we planned on using hot glue to create a water seal initially and then using epoxy, but due to time constraints we stuck with hot glue for our final prototype.

To secure the water pump in the water storage section, we needed to hot glue it to the bottom since the water pump only ejects water when completely submerged. Also, to refill the water, we created a hole in the top of the water storage area and added a water bottle cap that could be unscrewed to refill.

The last part of the water pump to be implemented was to create a nozzle to reduce the volume of water coming out of the hose. This was easily accomplished by using the nozzle from a syringe, cutting it to the proper length and width, and then attaching it to the end of the hose that exited the shell.

3.3 Shell Production

After creating the SolidWorks design of our shell, we had to find a 3D printer to bring our creation to life. We searched for a variety of different 3D printing options like private contractors and SLO Makerspace but struggled to find a convenient solution. It wasn't until we stumbled upon innovation sandbox that we really were able to get this print off the ground. Innovation sandbox is a free 3D printing service for students at Cal Poly that we heard about through Matthew Williams, a friend and ME student at Cal Poly. Innovation Sandbox helped us get our first printed shell within a week but sadly we were unable to prototype with it due to warping. A second shell needed to be constructed quickly and after a meeting in the middle of spring quarter with Tina, we discovered that the EE department had acquired a 3D printer. We used this new 3D printer to create our 2nd shell print, a process that was done over three days.

4) Final Integration

By observing the cross section of our product below, one can view how we integrated all internal components.

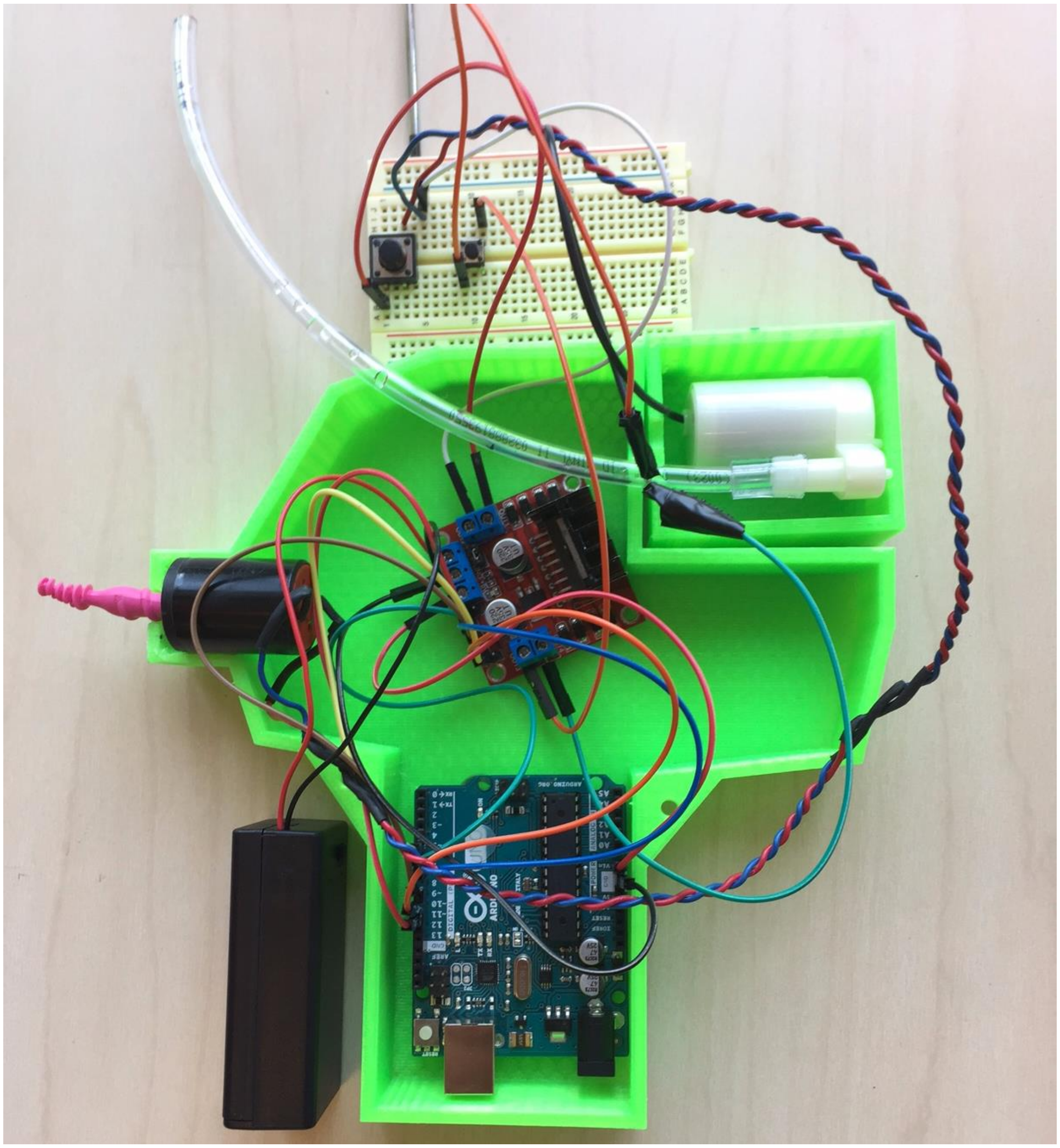

Figure 4.1: Cross Section of Component Layout / Prototype

Final integration of our components involved fitting all of the above components into our shell concisely while ensuring everything worked. To begin, we first placed all of the components within the shell and considered where they best fit. The cross section photo below shows our train of thought and how we imagined the placement to be.

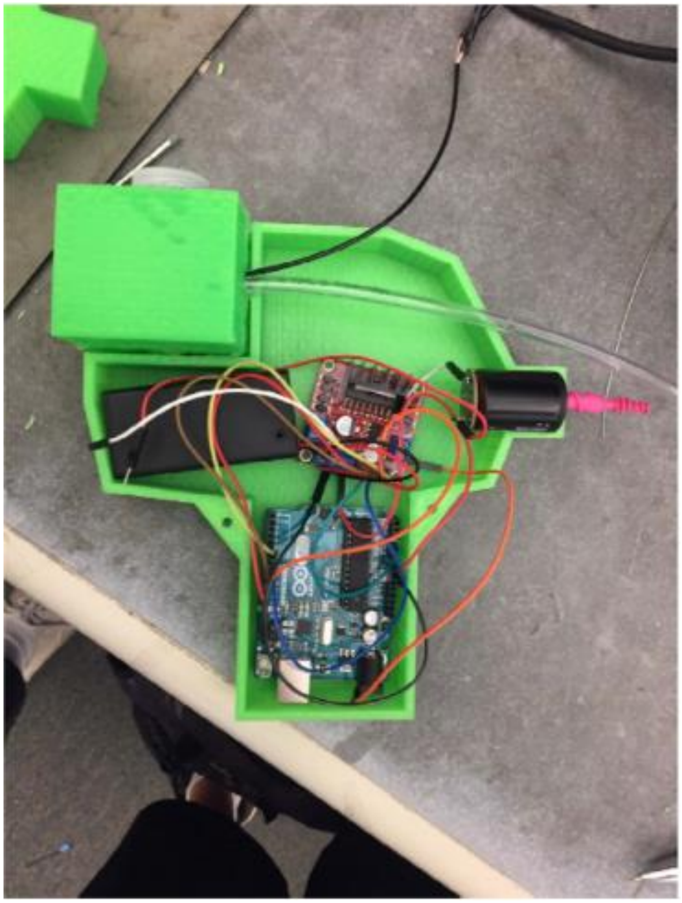

Figure 4.2: Finalized Prototype Placement Cross Section

The Arduino is housed in the bottom portion of the shell, as we desired based off of our initial shell design. The H-Bridge, 9 volt battery, and DC motor are all housed in the middle portion of the shell. Lastly, the water pump and hose are in the top portion of the shell with an easy to remove water storage design.

In order to successfully fit all components into our shell, we had to Dremel our 3D printed shell and make room for the motor and hose. The Dremel is a very powerful tool that requires absolute precision and accuracy upon use. One needs to make sure they measure twice and cut once because they Dremel will absolutely gnaw away at your material. A better solution to the Dremel would have been accounting for the holes we needed prior to printing by implementing them through SolidWorks. A 3D printer will make your holes very clean and accurate, reducing any need for post print processing. Initially, we wanted to print our first shell and then through working with it, adjust the 3D CAD file and print an updated shell that fit our specifications better. However, due to time constraints, we made the first 3D printed shell our final prototype.

The way that all the components in the design communicate with each other can be observed based off the Arduino schematic (Figure 4.3) below.

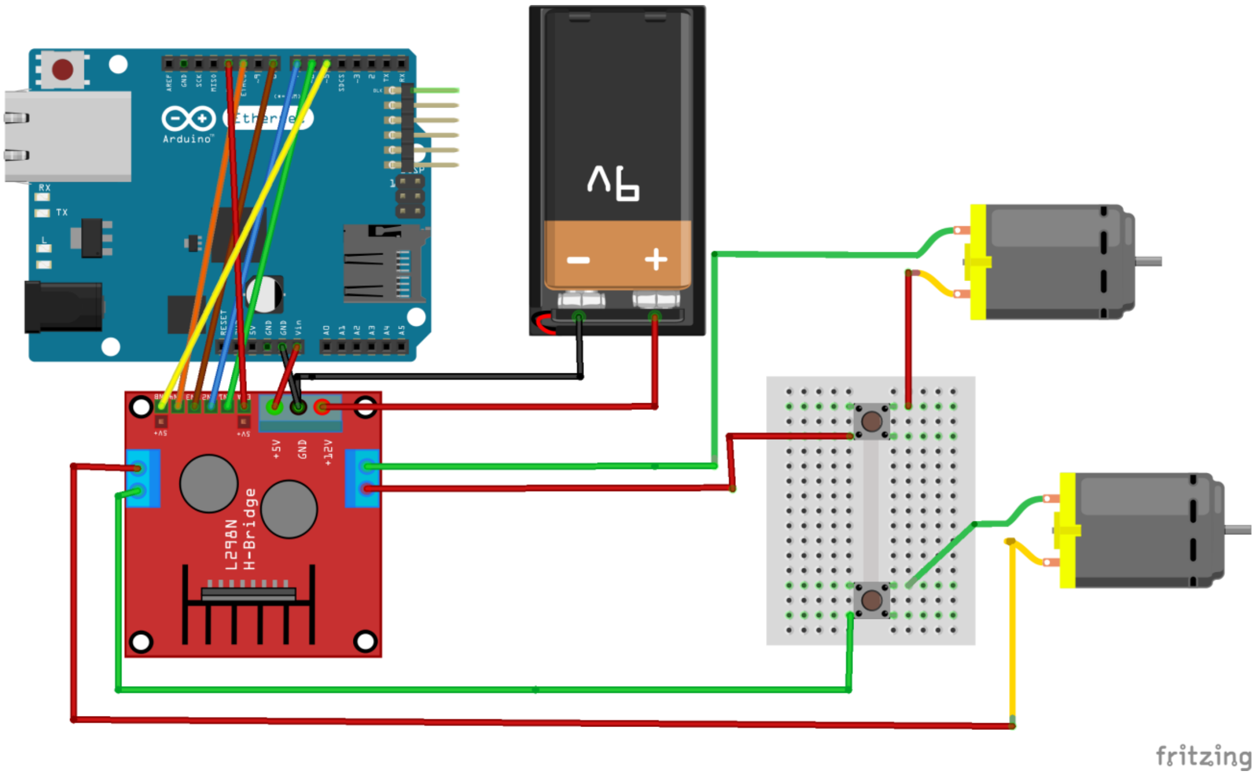

Figure 4.3: Arduino Interfacing Schematic

The 9V battery powers the H Bridge by delivering 9 Volts into the Vin of the L298N H-Bridge. The ground of the h-bridge has the ground of the 9 Volt battery going into it and another wire coming out of the L298N ground, creating a common ground to the Arduino 5 Volts output is delivered from the L298n H-bridge and used to power the Arduino. Next, we wire up the L298N H-Bridge to the Arduino so one can program both desired motors. This is done by hooking up ENA from the l298n to pin 11 on the Arduino, Input pins 1-4 on the L298n to output pins 6-10 on the Arduino, and EN B from l298n to pin 5 on the Arduino. After we wired up the h-bridge appropriately, we added the DC motors with button control to the L298n and were ready to roll. The buttons act as an open circuit while they are not being pressed. Upon pressing them, the circuits close and the motors begin receiving power.

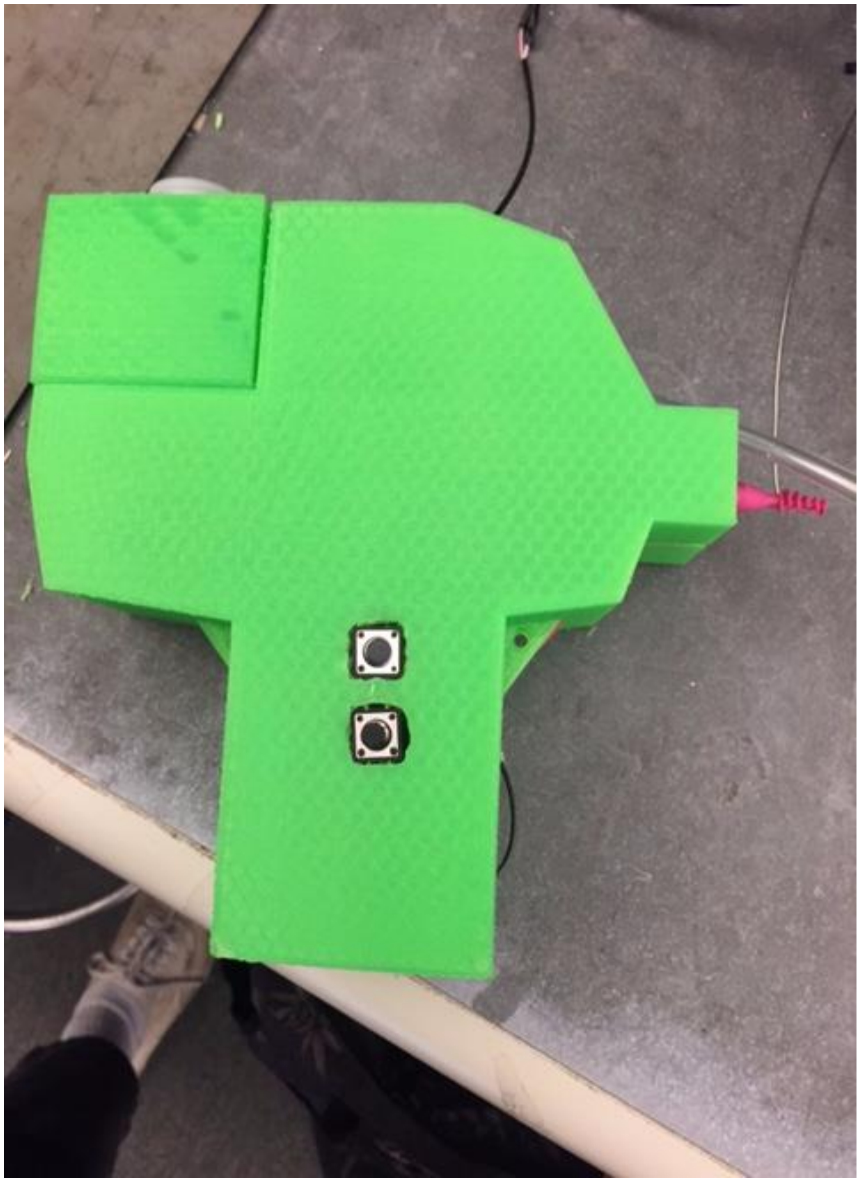

Figure 4.4: Closed and Finished Prototype

The final step required closing the shell. This intended to be accomplished with two screws to create the watertight seal, but the locations of the hole did not allow for a proper seal. A third screw hole on top potentially could've been able to accomplish the watertight seal, but with time constraints we ended up using hot glue to create the seal we needed for our prototype. Obviously, a non-prototype version would not include hot glue because of the fact that it needs to be openable to replace the battery after extended use.

5) Conclusion and Future Topics

Even though we consider this project successful in accomplishing its objective, we realize this is just a prototype. From our experiences in building this, we could have been able to build a smaller, more ergonomic shell, use cheaper components, and have better options with feedback for the user.

If we were to work on this again, we could try to use an IC since the most expensive portion of our project was the Arduino microcontroller. Through additional research and more complex circuitry, we could create an H-Bridge and a Pulse Width Modulation circuit to accomplish the same task with less area used and less cost.

Also, much of our project used empirical data for testing purposes. With more time and funding this project could analyze the effectiveness of different options and resources for our project such as finding the proper RPM for our scraper motor and water pump that fits the user the best. There could even be circuitry to allow the user to adjust both the motors with simple buttons, knobs, or switches.

Overall, we are satisfied that we accomplished what we set out to do. The function of the project was accomplished but with additional time we could flesh out other options to create this into a more user-friendly product that has viability in the open market.

Appendix A: Code for Arduino Uno Microcontroller

```
// connect motor controller pins to Arduino digital pins
// motor one
int enA = 11;
int in1 = 10;
int in2 = 8;
// motor two
int enB = 5;
int in3 = 7;
int in4 = 6;
void setup()
{
  // set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}
void demoOne()
{
  // this function will run the motors in both directions at a fixed speed
  // turn on motor A
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(enA, 200);
  // turn on motor B
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(enB, 200);
```

```
  delay(2000);
  // now change motor directions
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(2000);
  // now turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
void demoTwo()
{
  // this function will run the motors across the range of possible speeds
  // note that maximum speed is determined by the motor itself and the
operating voltage
  // the PWM values sent by analogWrite() are fractions of the maximum speed
possible
  // by your hardware
  // turn on motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  // accelerate from zero to maximum speed
  for (int i = 0; i < 256; i++)
  {
    analogWrite(enA, i);
    analogWrite(enB, i);
    delay(20);
  }
  // decelerate from maximum speed to zero
  for (int i = 255; i >= 0; --i)
```

```
  {
    analogWrite(enA, i);
    analogWrite(enB, i);
    delay(20);
  }
  // now turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
void loop()
{
  demoOne();
  delay(1000);
  demoTwo();
  delay(1000);
}
```