Electric Paintbrush Cleaner

ECE 445 Design Document

Luis Bibian, John Kriston, Yael Legaria
Team 65
TA: Madison Hedlund
February 27, 2020

1. Introduction

1.1 Objective

When painting, it is commonplace to have a cup of water used to clean the paintbrush when switching between colors. However, after only a handful of times of dipping the paintbrush in the water to clean it, the water becomes heavily stained with residual paint. This is problematic, as the unclean water will start seeping into the brush and lead to the unwanted mixing of colors on the canvas. Although having a designated cup of water to clean the paintbrush is a simple way to mitigate the degree of which the mixing of colors occurs, as the painter progresses in the painting process and continues to use the same cup of water, the effectiveness of the cleaning water only worsens, meaning that this solution isn't one that is fit for long-term painting sessions that require a diverse palette of colors.

Our proposed solution is to create a device that aims to clean the brush while eliminating the contamination of the cleaning water. This will be achieved by having a receptacle of clean water that will be sprayed onto the paintbrush upon its insertion, with any dirty water dripping into a separate receptacle to hold the dirty water. Furthermore, in order to address the buildup of paint particles that get caught between paintbrush bristles, we are also proposing a cleaning mechanism that consists of a motorized brush to clean between the paintbrush fibers. This will allow the painter to keep from getting their hands dirty as they usually would by using their fingers to rub off any residual paint stuck in the brush. In order to accomplish this task, our project will need to sense when the paintbrush has been inserted for cleaning, distribute the water to rinse the paintbrush, and then move the motorized cleaning brush through the paintbrush.

1.2 Background

The usual technique that painters resort to in order to correct the problem of contaminated water interfering with the quality of a painting is to either use two separate containers of water (one for getting most of the paint off the brush, and the other for rinsing the brush after the first clean), or to use a very large tupperware/bucket such that any paint that is rinsed into the water will achieve a higher degree of dilution due to the larger area of water. However, neither of these solutions actually fix the problem of the cleaning water getting contaminated - they instead only serve to slow down the process. Eventually, the painter will have to get up and change the water, which only interrupts their workflow.

It is because of these inconveniences that our solution aims to completely separate the clean water reservoir from the paintbrush during cleaning. Instead, clean water will be delivered to the brush via a waterpump and the resulting dirty water will be collected in a separate container. While this can also be achieved by running a paintbrush under the faucet, the problem is that it is not always convenient to have to paint in a kitchen where food is present or within the confines of a bathroom. Thus, our solution is one that doesn't restrict the painter to a particular area of the house that may be unfavorable.

1.3 Physical Design

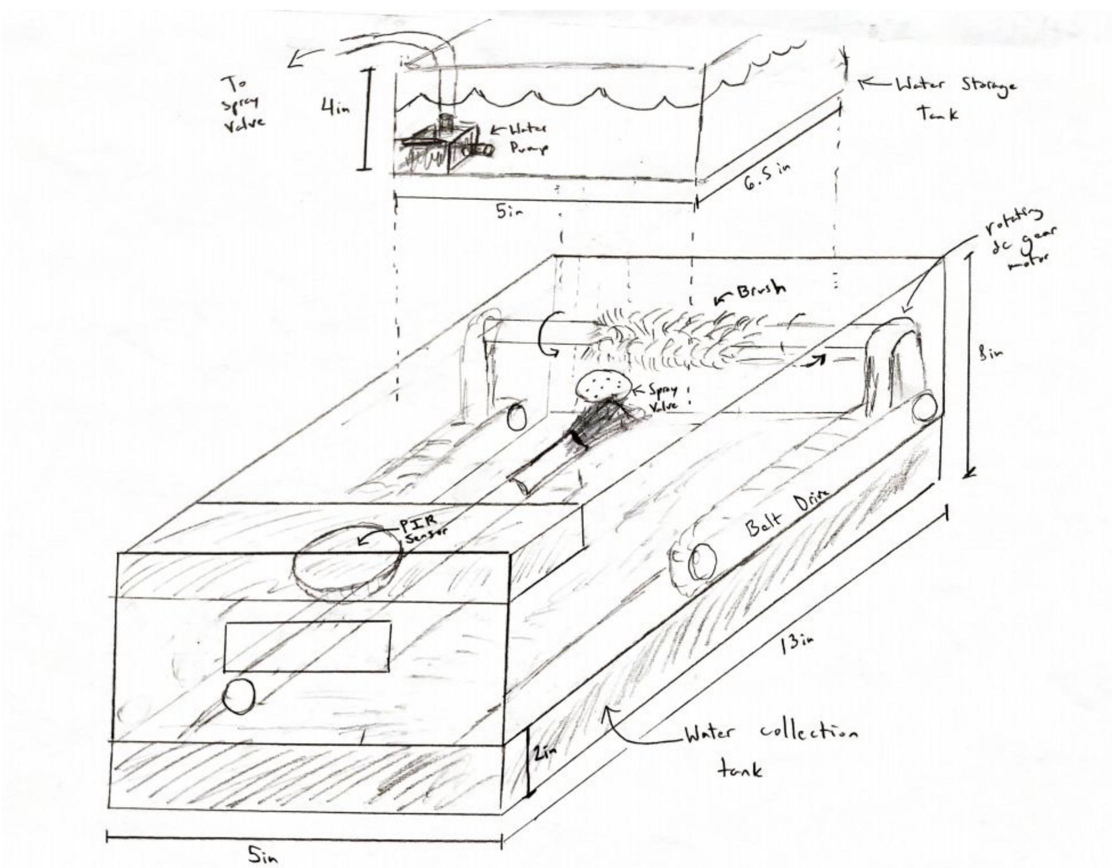

Figure 1: Physical Design

1.4 High-level Requirements

- The water from the cleaning tank should be successfully routed to the paintbrush to rinse it clean, with any used water being collected in a separate removable receptacle, for a minimum of four iterations.
- The motorized brush cleaning mechanism should be able to work any dried up paint particles out of the paintbrush, and should demonstrate a range of movement that allows it to move back and forth via the actuator system.
- The microcontroller must be able to process the incoming sensor data from the PIR sensor, while also sending the correct signals to the motor drivers and the dc water pump switch at the appropriate times .

2. Design

2.1 Block Diagram

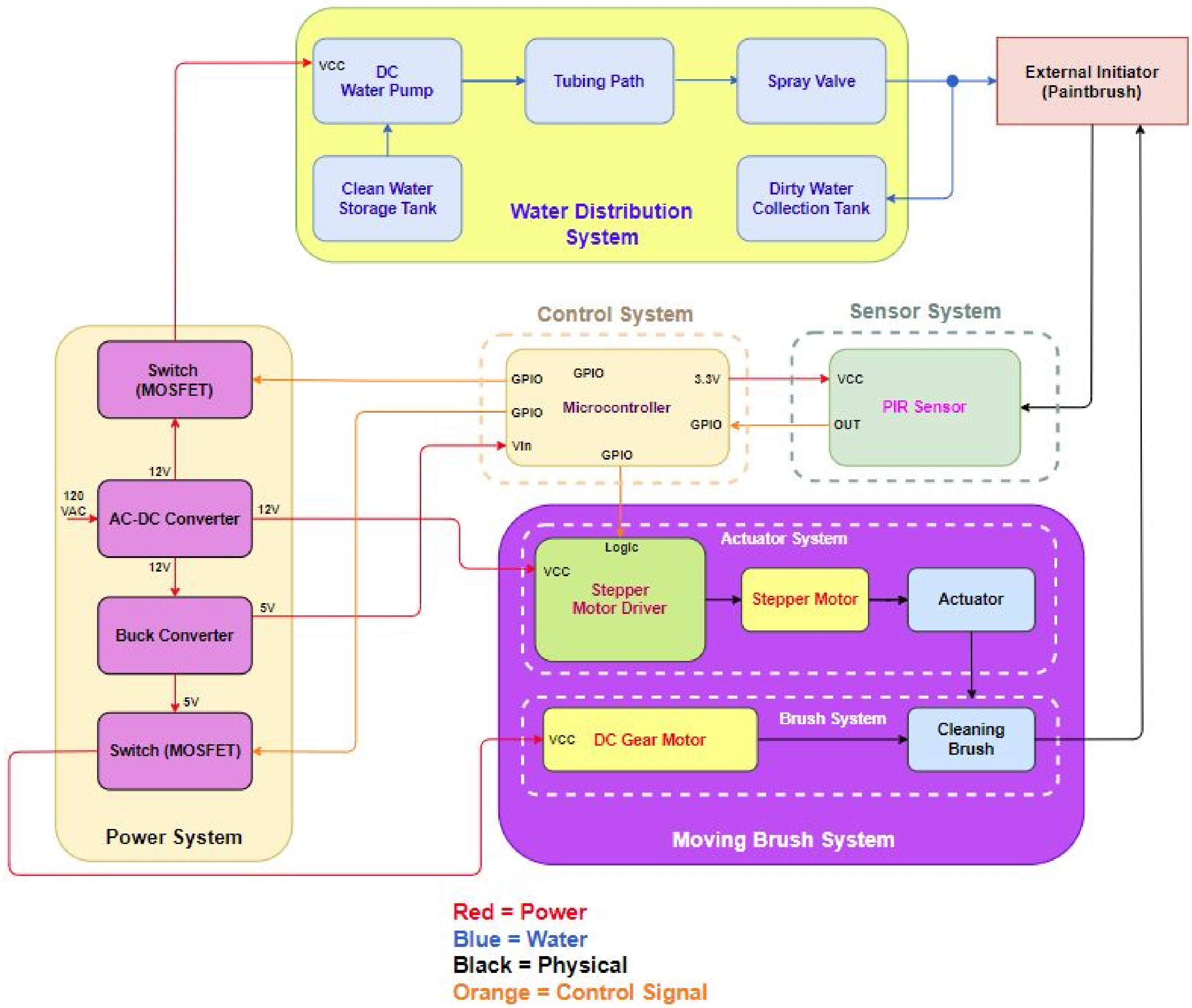

Figure 2: Block Diagram

The electric paint brush cleaner will begin its cleaning cycle once the PIR sensor detects movement from an inserted paintbrush. At this point, the microcontroller will process the data coming from the triggered PIR sensor and begin to initiate the cleaning cycle. The microcontroller will first switch on the MOSFET to allow the dc water pump to turn on and begin pulling clean water from the designated clean water storage tank. This water is then routed through a tubing path and exits through a spray valve to rinse the paintbrush, with the dirty water being collected in a separate tank below. Once the rinsing portion of the cycle is over, the microcontroller will then send the appropriate signals over to the stepper motor driver to move the brush system forward such that the cleaning brush will be able to start cleaning the paintbrush bristles. Once the actuator has moved the cleaning brush system forward, the microcontroller then starts to rotate the cleaning brush attached to the dc gear motor via its corresponding driver. Once this process is over, the microcontroller will retract the brush system further back by driving the actuator system once again, signaling the end of the cleaning cycle.

2.2 Control System

The control system behaves as the central interface of the electric paintbrush cleaner and is responsible for sending the proper signals to the other subsystems at the appropriate time. The control system also takes in input data from the sensor system to determine when to enter the cleaning process.

2.2.1 Microcontroller

The microcontroller is powered from the 5V output of the power system's ac-dc converter. Upon receiving a high signal from the PIR sensor, the microcontroller will then supply the gate of the dc water pump switch with 5V via one of its digital output pins to begin routing water to the paintbrush. The microcontroller will then wait for a few seconds to allow the water enough time to move through the system before sending control signals to the stepper motor driver such that the actuator can move the cleaning brush forward. After this, the microcontroller will supply the gate of the dc gear motor switch with 5V to allow the cleaning brush to rotate. The microcontroller will once again let a few seconds pass before sending control signals to the stepper motor to retract the actuator, signalling the end of the cleaning cycle. The microcontroller is also responsible for keeping track of the timing of the cleaning cycle such that any triggering from the PIR sensor while the cleaning cycle is in progress won't restart the cycle.

Requirement	Verification
Microcontroller must be able to read the incoming signal from the PIR sensor.	A. Power PIR sensor with 3.3V from Arduino and connect digital pin 2 to OUT of PIR sensor. B. Write Arduino sketch to turn on on-board LED whenever digital pin 2 reads high.
Microcontroller must send appropriate signals to MOSFETs and the stepper motor driver.	A. Check appropriate signals are being sent by viewing them on oscilloscope at this time
Microcontroller must demonstrate accurate timing such that every cleaning cycle completes within 1 second of average estimated execution time.	A. Connect microcontroller digital pins to appropriate devices as detailed on schematic. B. Write Arduino sketch to send signals through pins at the appropriate times, and have microcontroller start a timer once the high signal has been read from PIR sensor. C. Print execution time at the end of the cycle to the Serial Monitor.

2.3 Sensor System

The sensor system (consisting of the PIR sensor) is a fundamental part of the design that allows the user to clean the paintbrush in a hands-free manner. The triggering of the cleaning cycle depends on the sensor system properly detecting when a paintbrush is inserted, at which point the PIR sensor will send a signal to the microcontroller, indicating that water can start being directed to the paintbrush.

2.3.1 PIR Sensor

For our design, it made the most sense to choose a Passive Infrared (PIR) sensor as our method for detecting the insertion of a paintbrush into the cleaning device. A device that is similar in principle to the construction of our electric paintbrush cleaner is a pencil sharpener, which uses a button that is pushed by the tip of the pencil to detect when it has been inserted - however, using the same amount of force to try to push a button with a paintbrush would damage the bristles, and would also result in paint getting stuck to the button, which is why the PIR sensor was chosen instead. The PIR sensor can be directly powered from the 3.3V output of the microcontroller. The PIR sensor should only detect movements that are internal to the device (i.e. external movements such as someone walking by the device or moving their hand in front of the device shouldn't trigger the sensor).

Requirement	Verification
Sensor transmits a signal to the microcontroller upon the detection of paintbrush inserted into device.	A. Power PIR sensor with 3.3V from Arduino and measure voltage across GND and OUT on oscilloscope. B. Insert a paintbrush below the sensor in a linear movement and verify that the signal goes high on oscilloscope.
Sensor is sheltered enough to prevent triggering from any external movement that takes place outside of the device.	A. Place PIR sensor in a compartment such that a portion of it is facing downwards through a detection slot, with the rest of the sensor being blocked off by the compartment. B. Power the PIR sensor via Arduino, with OUT connected to Arduino digital pin 2. C. Write Arduino sketch to turn on on-board LED whenever digital pin 2 reads high. D. Simulate walking or any other physical movement that is perpendicular to the placement of the sensor. LED should remain off.

2.4 Power System

The power system is responsible for supplying power to the appropriate modules in the electric paintbrush cleaner. The power system consists of an ac-dc converter to convert ac from the wall to 12V dc, a buck converter to step the 12V dc down to 5V, and two MOSFETs that can be switched on and off to control the flow of power from the ac-dc converter and buck converter to the motors.

2.4.1 AC-DC Converter

The device should be able to run off ac power so that it can be plugged into a wall outlet.
A 120V ac input is converted to 12V dc that can be supplied to the stepper motor driver and the dc water pump. This 12V will also be supplied to the buck converter such that it can be stepped down to 5V. Since the ac-dc converter is our main power source for all of the devices, the converter must be able to supply sufficient current, with estimates of around 1A for the microcontroller, 0.2A for the dc gear motor, 4A for the stepper motor driver, and 1A for the dc water pump, totalling 5.2A.

2.4.2 Buck Converter

The buck converter steps down the 12V from the ac-dc converter to a usable 5V for the microcontroller and the dc gear motor. Using a step-down switching converter is the cheaper alternative compared to using a dual-output power supply that can meet the current requirements of the project. The current available at the output of the buck converter must also be able to meet the current demands for the dc gear motor and the microcontroller, which comes out to be an estimated 1.2A. Our buck converter of choice is the TI PTH08080W, which accepts an input range of 4.5V-18V and provides an output current of up to 2.25A [1].

Requirement	Verification
The ac-dc converter must provide 12V within ± 5%, while supplying at least 6A of continuous current.	A. Measure output voltage of ac-dc converter with oscilloscope, making sure voltage remains between 11.4V and 12.6V. B. Use a multimeter to measure the current while an electronic load is configured to draw 6A of current.

The buck converter output must provide 5V within +5%, while supplying at least 1.5A of continuous current.	A. Measure output voltage of buck converter with oscilloscope, making sure voltage remains between 4.75V and 5.25V. B. Use a multimeter to measure the current while an electronic load is configured to draw 1.5A of current.

2.4.3 MOSFETs

The MOSFETs are used as simple switching mechanisms that allow the microcontroller to turn the dc gear motor and dc water pump on and off at set times. Since the microcontroller can only source a limited amount of current, the MOSFETs will be used as a switch in order to drive the dc water pump and dc gear motor directly from the power system by connecting them to the dc outputs of the power system. The MOSFETs will be used in a low-side configuration in order to allow the microcontroller to switch the MOSFETs without the need for a gate driver.

Requirement	Verification
MOSFET drain-source channel must conduct upon the application of voltage to the gate via the microcontroller.	A. Connect MOSFET Drain to one terminal of a resistor, with the other terminal of the resistor connected to power supply. B. Attach a 10kΩ pull-down resistor to Gate of MOSFET, tying it to GND. Connect Source of MOSFET to GND. D. Configure oscilloscope to measure current into Drain of MOSFET. E. Apply a voltage greater than threshold voltage from microcontroller digital output pin to transistor Gate, verifying that current is conducting from drain to source on oscilloscope.

Voltage drop across MOSFET drain-source terminals must be less than 0.2V when it is conducting.	A. After performing previous verification, while the MOSFET is still conducting, place a voltmeter across the drain and source terminals of the MOSFET and measure the voltage drop.

2.5 Water Distribution System

The water distribution system controls the storage, flow, and collection of water throughout the electric paintbrush cleaner. This system consists of a water storage tank that sits on top of the device and holds the clean water, a dc water pump that routes water from the storage tank to the spray valve through a tubing path, and a dirty water collection tank that sits at the bottom of the device and can be easily removed. The purpose of this system is to facilitate the cleaning of the paintbrush in a way that prevents the clean water from being contaminated by any paint.

2.5.1 Water Tanks

The clean water storage tank holds the water that will be used to rinse off the paintbrush. This tank will sit on top of the physical cleaning compartment where the brush is inserted into. Meanwhile, the dirty water collection tank is responsible for collecting the used water that is being sprayed down onto the paintbrush. This compartment will sit at the bottom of the device and can be pulled out for cleaning. Both tanks should be separately removable by the user in order to easily replace/empty water.

Requirement	Verification
Both tanks should be easily removable without spilling any water (at least three cups or 43.3 cubic inches of clearance between water level and top of container).	A. Fill up the water tanks to the designated water level and carry them and move them around to make sure no water overflows.

2.5.2 DC Water Pump

The dc water pump is responsible for taking the water from the clean water storage tank and delivering it to the paintbrush via the tubing path and spray valve. The dc water pump runs on 12V supplied from the ac-dc converter and sits within the clean water storage tank. In order to make efficient use of the clean water, the dc water pump will be turned on in short bursts, with the spray valve increasing the pressure of the water delivered to the paintbrush.

Requirement	Verification
The dc water pump should deliver water to the cleaning mechanism/brush-head at a minimum rate of 7 L/min.	A. Fill a container/tupperware with 3 cups of water and make a mark to indicate the water level (3 cups of water is approximately 0.71 Liters). B. Run the dc water pump for 6 seconds. If the water in the container is above the 3-cup water-level mark then the water pump is achieving the requirement of pumping water at 7L/min.

2.6 Brush System

The brush system consists of a cleaning brush attached to a rod that spins via the dc gear motor. This system is physically situated on top of the moving actuator such that it can move back and forth within the device.

2.6.1 DC Gear Motor

The dc gear motor will be used to rotate the cleaning brush in a direction parallel to the paintbrush bristles to assist in the cleaning process and work any dried-up paint particles out of the paintbrush. A dc gear motor was chosen because of its relatively small size, along with its ability to provide enough torque to turn a small cleaning brush at a speed that won’t damage the paintbrush bristles. The dc gear motor is powered by the 5V available at the output of the buck converter.

Requirement	Verification
Dc gear motor speed must be between 100 RPM and 150 RPM to prevent damaging the paintbrush while still effectively cleaning it.	A. Using a 5V supply, form a voltage divider using a 10kΩ resistor and a photoresistor. B. Set up an Arduino to measure the voltage across the photoresistor and aim a laser pointer at the photoresistor. C. Attach a black piece of tape to the dc gear motor rod that will interrupt the laser beam upon rotation. D. For the duration of a minute, use the Arduino to keep track of the number of times that the voltage across the photoresistor rises above 4V (indicating that the beam has been interrupted) to get a measure of the RPM.

2.7 Actuator System

The actuator system consists of an actuator, a stepper motor, and a stepper motor driver, and is responsible for moving the brush system back and forth within the device by having the stepper motor rotate to extend and retract the actuator.

2.7.1 Stepper Motor Driver & Stepper Motor

The stepper motor driver is responsible for driving the stepper motor via signals from the microcontroller, and is powered with 12V via the ac-dc converter output. The stepper motor's rotational movement is used to extend and retract the actuator. A stepper motor was chosen specifically for this application because we need a motor that can provide enough torque to move the actuator without compromising on the accuracy and precision of the positioning over multiple cleaning cycles. For driving the stepper motor, the TB6600 driver [2] was chosen for its simple layout scheme and ability to drive a two-phase stepper motor with a stepping angle of 1.8° which is suitable for our design.

Requirement	Verification
Stepper motor rotation must consistently extend the actuator to minimum and minimum positions (rotor positioned at 0° and 180°) within 5° of accuracy.	A. Attach stepper motor to actuator. B. Manually position actuator such that it is fully retracted. C. Run Arduino sketch to rotate stepper motor by 180° fifteen consecutive times (simulating the movement from). D. The actuator should remain fully extended by the end of the code execution. Using a protractor and a bubble level, confirm that the rotor position is within 5° of the horizontal level.

2.7.2 Actuator

The actuator sits above the dirty water tank and is extended forward to bring the brush mechanism over to the paintbrush, and extended backward to retract the brush mechanism to allow the clean water to make contact with the brush for rinsing. The actuator position is changed by having the rotational movement of the stepper motor translated to linear movement via reciprocating motion.

Requirement	Verification
When fully extended, the actuator should position the cleaning brush such that the rod is within 3cm of the edge of the paintbrush bristles.	A. Drive stepper motor to extend actuator to maximum forward position. B. Drive dc gear motor and verify that bristles on cleaning brush are coming into contact with paintbrush. C. Use a ruler to measure from the metal rod to the tip of the paintbrush bristles and confirm that the gap between the two is no larger than 3cm.

When fully retracted, the actuator should position the cleaning brush far enough such that there is at least a 2cm distance between the edge of the spray valve and the nearest cleaning brush tip.	A. Drive stepper motor to extend actuator to position farthest away from the front of the device housing. B. Drive dc water pump to direct water through the spray valve and confirm that water is making contact with paintbrush without interference. C. Use a ruler to measure from the tip of the cleaning brush to the edge of the spray valve and confirm that there is at least a 2cm gap between the two.

2.8 Tolerance Analysis

The most critical overall feature is the water distribution system. This is the most critical because if the water distribution does not successfully dispense water at the appropriate times, no brushes can be cleaned. In addition, if water was not to be contained within the tubing path, it can damage the other components of our system. As a result, our water will have to be dispensed at a medium rate, and make sure to be collected in full.

We hypothesize that each brush will take roughly 10 seconds (.167 minutes) to be sprayed by the spray valve. The selected 12 V DC water pump can dispense water at a rate of 550 L/hour, or 9.16 L/minute at maximum voltage [3]. Therefore, we expect that 7 L of water in our initial clean water storage tank should be able to last us just over 45 seconds of cleaning, cleaning 4 brushes in total. As we expect minimal loss of water in between cleaning brushes, we therefore can calculate and show

$$(1)\ (7\ L\ of\ Water)/(9.16\ L\ of\ water/minute)\ =\ .764\ minutes\ of\ cleaning$$

$$(2)\ \ 7\ L\ of\ Water\ -\ (9.16\ L\ of\ water/minute)(.167\ minute/brush)\ (4\ brushes)\ =\ .881\ L\ remaining$$

We will look at the dirty water storage tank at the end of the cleaning cycles to get the exact measurement of how much water we used, and we can also look at our initial storage tank to see how much water we have left. This is essential so we know we are cleaning our brushes efficiently and optimally so we don't run out of water. The success of this shall depend on the rate of water dispensed and the tubing path since this will also determine the flow rate.

Furthermore, the worst case error is that water is pumped at a rate too fast for even one paintbrush to be fully cleaned. We estimate this can only happen if water is pumped at a rate of higher than 41.916 L/minute. As shown by this calculation, the worst case is that:

$$(3)\ \ 7\ L\ of\ Water\ -\ (>41.916\ L\ of\ water/minute)(.167\ minute/brush)(1\ brush)\ =\ 0\ L\ remaining$$

Obviously, this worst case error is very unlikely because we are working at maximum voltage for our DC pump, and so there would have to be a component error for this to occur.

2.9 Schematic

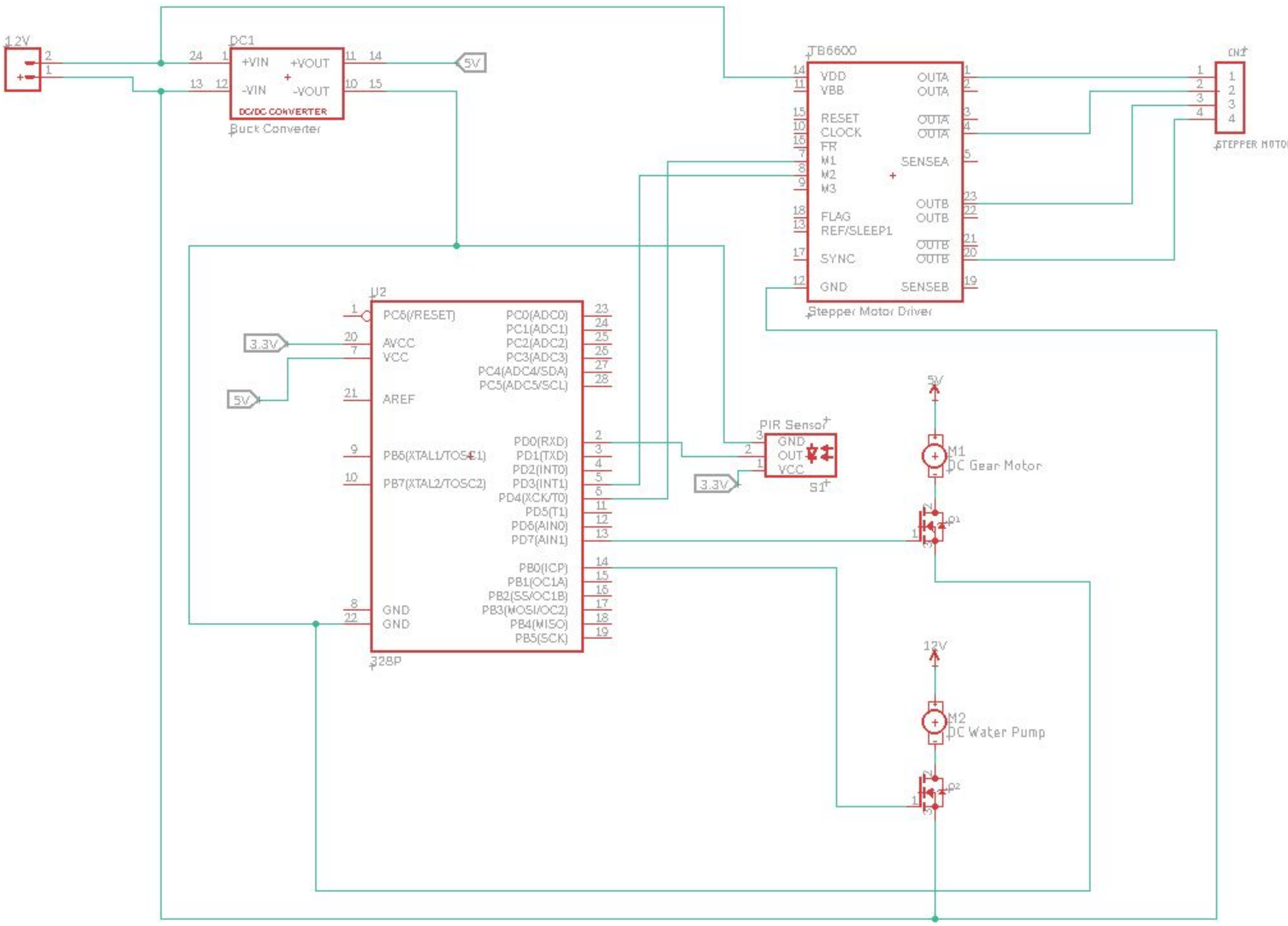

Figure 3: Schematic Design

3. Cost Analysis and Schedule

3.1 Cost Analysis

3.1.1 Cost of Labor

Assuming that each of us works 10 hours per week at an hourly rate of $35/hour over the course of 10 weeks, the total cost comes out to be:

(4) $$3 \cdot (\frac{\$35}{hour}) \cdot (\frac{10\ hours}{week}) \cdot 10\ weeks \cdot 2.5 = \$26,250$$

3.1.2 Cost of Parts

Description	Manufacturer	Part #	Quantity	Unit Price
MOSFET	Nexperia	PSMN022-30PL	2	$0.90
Buck Converter	Texas Instruments	PTH08080WAH	1	$7.62
Stepper Motor Driver	Sorotec	TB6600	1	$13.59
Stepper Motor	Usongshine	17HS4401	1	$8.98
DC Water Pump	DFRobot	FIT0563	1	$9.29
DC Gear Motor	Antrader	N/A	1	$10.99
AC-DC Converter	Alito	ALT-1208	1	$18.99
PIR Sensor	Radioshack	2760347	1	$11.90
Microcontroller	Microchip Technology / Atmel	ATMEGA328P	1	$2.08
Misc. PCB and Manufacturing	Various	Various	N/A	$25.00
Total Parts Cost				**$110.24**

Table 1: Cost of Parts

3.1.3 Total Costs

Item	Associated Cost
Labor	$26,250
Parts	$110.24
Total	**$26,360.24**

Table 2: Total Cost

3.2 Schedule

Week	Luis	John	Yael
3/2	Finalize and order necessary parts	Talk to machine shop regarding physical design of device	Start 3D modeling of device's outer construction
3/19	Start verifications for power and brush system	Start verifications for water distribution system	Finalize PCB Design for early order
3/16	Begin assembling physical construction of brush system	Refine measurements needed for proper flow of water throughout distribution system	Start writing control system code and verify working sensor system
3/23	Start combining brush system, actuator system, and outer device shell	Test control code with each subsystem	Test actuator system
3/30	Integrate subsystem with full system	Integrate subsystem with full system	Integrate subsystem with full system
4/6	Update brush system	Update water	Update physical

	if needed	distribution system if needed	construction and actuator system if needed
4/13	Start final paper	Start extra credit poster	Start final presentation
4/20	Prepare for mock presentation and final demonstration	Prepare for mock presentation and final demonstration	Prepare for mock presentation and final demonstration
4/27	Complete final presentation, final paper, and poster	Complete final presentation, final paper, and poster	Complete final presentation, final paper, and poster

Table 3: Weekly Schedule

4. Ethics and Safety

In order to enforce the IEEE code of ethics, specifically, section 7.8.1, the group acknowledges that an obvious safety issue which can arise with this project is the close proximity of water to electronics [4]. The group shall take the necessary safety precautions during project development to prevent the risk of electrical shock including the use of GFCI outlets, appropriate grounding of all electrical components, and proper isolation of all water. The group will also make sure of waterproofing all circuitry, as well as, creating a design durable enough to prevent spillage of water onto the electronics in case of an accident. Furthermore, the group has decided to develop a safety manual that shall inform users of the potential risks associated with using the appliance, as well as, the precautions they should take to minimize these risks. These precautions include, but are not limited to, not exceeding the max water level of the clean water tank, always emptying out the bottom collection tank before use, and always ensuring that the water system is working smoothly without any clogging. This safety manual should closely resemble those which are distributed with similar water appliances such as in-home decorative water fountains.

Realistically, we must accept that there could be flaws with our final product, and understand that it is important to be honest with our claims as is stated in section 7.8.3 of the IEEE code of ethics [4]. This means that if in reality our product is unable to clean a paintbrush as well as other common alternatives, we must be honest about its effectiveness. Also, if we believe that our product is effective in the end then, we are responsible for proving its effectiveness with sufficient data.

Furthermore, our group will follow the IEEE code of ethics, section 7.8.7. This section explains "how we must seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others [4]." In particular, our group is going to use established and outside technologies to build our paintbrush cleaning system, and so we will cite which parts we used and where we got them from. As of right now, this includes a MOSFET and a stepper motor plus driver. In addition, our group will acknowledge and rectify any design flaws as we assemble our system. It's imperative to acknowledge when a part or subsystem is not quite working the way we intended, and to replace or fix it immediately. This will allow us to create a better functioning system, and is important for the safety of us and all our peers.

5. Citations

[1] Texas Instruments, “2.25-A, WIDE-INPUT ADJUSTABLE SWITCHING REGULATOR”, [Online]. Available: http://www.ti.com/lit/ds/symlink/pth08080w.pdf [Accessed: 20-Feb-2020].

[2] Sorotec, “TB6600 Stepper Motor Driver”, [Online]. Available: https://www.mcielectronics.cl/website_MCI/static/documents/TB6600_data_sheet.pdf [Accessed: 20-Feb-2020].

[3] DFRobot, “DC Mini Immersible Water Pump (6V~12V) ,” FIT0563, [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/FIT0563_Web.pdf [Accessed: 20-Feb-2020].

[4] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 20- Feb- 2020].

[5] Nexperia, “PSMN022-30PL” FIT0563, [Online]. Available: https://www.mouser.com/datasheet/2/916/PSMN022-30PL-1600234.pdf [Accessed: 20-Feb-2020].

[6] RadioShack, “User’s Guide PIR Sensor” [Online]. Available: https://github.com/RadioShackCorp/2760347-PIR-Sensor/blob/master/User's%20guides/2760347_PM_EN.pdf [Accessed: 20-Feb-2020].