Design for a Toilet Appliance that Automatically Closes the Toilet Lid before Flushing

A Major Qualifying Project Report

Submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the Requirements for the

Degree of Bachelor of Science in Mechanical Engineering

Submitted to Project Advisor

Professor Jianyu Liang, Worcester Polytechnic Institute

Submitted by

Agafiya Konnova

Faris A. Shaikh

Yin Zhang

Kunjun Zou

Date: 14 August 2020

Abstract

Across the globe, differing bacteria and viruses are resulting in thousands of deaths. Some of these harmful bacteria and viruses are aerosolized from toilet bowls. This project team has embarked upon a journey to develop a smart toilet appliance to address the issue of airborne bacterial transmission from toilet bowls. This appliance will limit the spread of harmful bacteria and viruses by closing the seat lid before they are ejected into the air during flushing. In this design, two electric motors (a Brushed DC Motor and a Stepper Motor) along with two sensors (a Button Sensor and an Ultrasonic Sensor) operate in unison with an Arduino microcontroller to create the smart toilet appliance. This appliance is designed to be affordable and easy to install.

Acknowledgments

We would like to thank Professor Jianyu Liang for assembling and inducting us into this Major Qualifying Project. We are grateful to the Teaching Assistant Zhaolong Zhang and Leo Gross, for their technical assistance and support. Without their principal guidance, we could not have accomplished as much within the timeframe of the summer academic terms. Lastly, we would like to thank all of the remaining supporters who aided and encouraged through these trying times during this pandemic. These individuals helped us tackle the challenges that arose while working on the circuits, CAD drawings, or more. This project has been a labor of love, and we hope that some benefit comes of it.

Table of Content

Table of Figures

Figure Title

Table of Tables

Table Title

Executive Summary

Today, across the globe, the death toll of the latest novel Coronavirus has exceeded approximately 150,000 people, as seen in Figure 1. Any measures that can save lives should be immediately implemented. This project team has devised a smart toilet appliance that can prevent the spread of harmful bacteria and viruses, such as COVID-19, from spreading into shared spaces.

The project was divided into two primary goals. The first goal was focused on designing a smart toilet appliance prototype that closes the toilet lid before flushing away the waste. The second was to create casings for all of the design's components. To accomplish these goals, the team began researching previous smart toilet designs and comparing different motors and sensors that could be used in the project. Additionally, features and design architecture customers might appreciate as well as how to protect the components were considered. The design decisions included the use of stepper and brushed DC motors to move the lid and flush the toilet, and ultrasonic and button sensors to inform the Arduino and its Arduino Shield. The team developed a computer-aided design (CAD) model to illustrate the design features and operation of the appliance using the dimensions that were measured and the technical solutions that were developed.

The design places the motors and sensors in the most efficient locations possible. The electrical components on the outside surfaces had been installed on the toilet to keep installation costs to a minimum. Modifications, such as drilling holes, would compromise the toilet's porcelain structure. Placing electrical components inside vessels intended to contain water would add complications such as developing adequate watertight seals and waterproofing components.

The compromising locations of the components have also been one of the design's major limitations.

The region surrounding commercial and residential toilets is often moister than in other spaces in a building; such an environment has limited the potential of the design. Components located in these wet regions must be resilient in moist environments. While the design includes covers that are meant to keep moisture away from the electronics, the sensors remain exposed to the environment. The sensors' exposure may decrease the product's lifespan. Testing long term durability is not feasible in this project and thus is outside the scope of this MQP. In the future, development of a fully commercializable product with a long-term service life may be further addressed.

1.0 Introduction

Coronaviruses are a family of viruses that can cause respiratory, enteric, hepatic, and neurological diseases among animals and humans [1]. It was first isolated in chicken embryos and identified in 1937. Not until 2002/2003, when the emergence of the Severe Acute Respiratory Syndrome (SARS) killed 774 people all around the world [2], did coronaviruses become a great concern for human health. At the end of 2019, the first case of Coronavirus disease 2019 (COVID-19) was identified in Wuhan, China [3]. Global attention recentered on the new Coronavirus as cases rose, Figure 1. This disease primarily spreads via droplets from nose or mouth when coughing or speaking. It remains possible to contract Conovirus through contact with surfaces covered with virus droplets, which then settle upon exposed skin, such as the face [4]. This disease raises awareness of the need to remain vigilant in spaces where the disease can easily spread. A toilet is an unsafe place. The generation of infectious aerosols from the toilet is the most common cause of disease transmission, similar to coughing, and sneezing, use of the toilet is inevitable. Therefore, it is important to ascertain techniques to avoid carrying the virus outside the toilet stall. Despite the uniqueness of COVID-19, persistent handwashing remains an effective method to stay healthy. Avoiding any interaction with material on top of toilet surfaces successfully lowers the possibility of contracting the virus.

The project aims to design a peripheral device for toilets to automatically close the toilet seat lid before flushing to reduce the interaction with the contagious toilet surfaces where different kinds of bacteria and viruses often exist. Flushing without first closing the lid will result in bacteria and viruses contaminating the entire lavatory [5]. An auto-closure cover is a convenient method for preventing the spread of these bacteria and viruses. To accomplish this goal, the project team has conducted research on various types of bacteria and viruses that exist in toilets, different types of toilets currently in use, and similar existing technology solutions to achieve a concrete understanding of the immense scale of the problem. After that, the team developed a blueprint of their own design and researched different characteristics of similar technological solutions, including motor selection, circuit construction, Arduino software development, sensor selection, and prototype assembly, to achieve the deliverables.

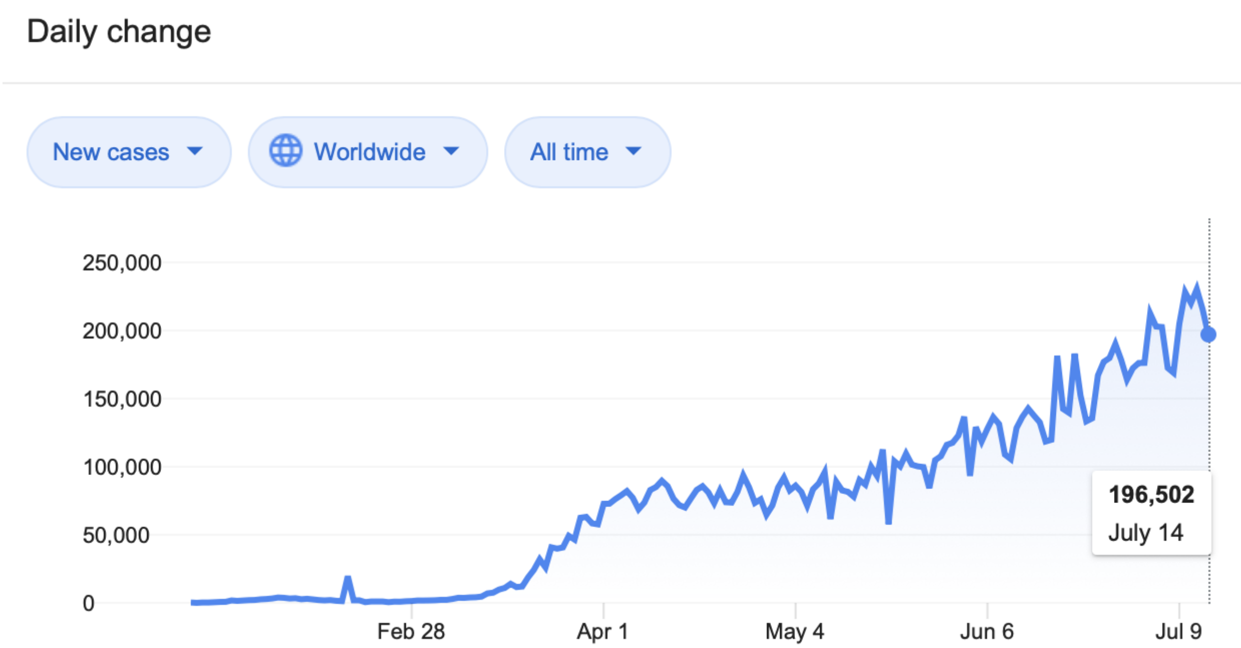


Figure 1: COVID-19 International Death Toll in 2020 [6]

1.1 Problem Statement

This project team seeks to develop a cost-effective solution to prevent the spread of harmful bacteria and viruses from toilet bowls. In the age of COVID-19, the smart toilet appliance will help limit the spread of this catastrophic virus by employing sensors that trigger automatic closure of the toilet lid before the waste is flushed. This appliance acts as an attachable module to existing domestic and commercial toilets.

1.2 Main Objective

This smart appliance employs the cost-effective components present in an Arduino Circuit Starter-Kit to develop a battery-operated smart toilet module that limits the spread of harmful bacteria and viruses by automatically shutting the toilet lid before flushing away the waste. When the toilet lid is closed, harmful bacteria can no longer be ejected from the toilet bowl.

2.0 Background

As a prerequisite to any designs, it is important to understand the history of the disease transmission space, toilets, and its functions currently work. To accomplish this objective, the project team researched and examined different toilet designs and solutions to toilet borne diseases.

2.1 The Potential Dangers of Flushing a Toilet

Toilet bowls are dirty places that contain hazardous bacteria and viruses that infect humans when gushing water ejects them out of the toilet bowl. A study conducted by Aston University determined that the flushing action led to the contamination of the entire room. After a single flush, the Serratia bacteria concentration in the air can increase from 13 CFU m^3 to 1370 CFU m^3, meaning that there are now one thousand times more bacteria and viruses units aerosolized, all of which are dangerous, that a person can ingest. Those very same bacteria and viruses can stick to the many surfaces of the lavatory for more than 60 minutes [7].

2.2 COVID-19 Can Remain on Surfaces

The SARS-CoV-2 virus, which led to the COVID-19, can remain on surfaces for days. According to the study conducted by the National Institutes of Health (NIH), the SARS-CoV-2 virus can survive on plastic and stainless-steel surfaces for up to three days. The virus, in its aerosol form, which represents the primary route of transmission, can stay alive in an enclosed space for up to days, increasing the risk of exposure and infection [8].

2.3 Toilet Seat

Upon the surface of the standard household toilet seat lies an average of 50 bacteria per square inch on a toilet seat [9]. Streptococcus is a common bacterium that is often found in the human throat. Strep throat and bronchial pneumonia are the results of Streptococcus' presence. Streptococci may also cause contagious skin infections, including impetigo (a type of rash that targets pre-school children and infants). Streptococci can also cause an invasive, serious skin infection called necrotizing fasciitis (also known as "flesh-eating" bacteria). Worst of all, about 39 percent of toilet seats harbor this menacing bug [9].

There is a chance that there are hordes of flesh-eating bacteria lying in wait to strike victims lying on a public toilet seat, such an infection from a shared seat, it is highly unlikely. Only about 1 percent of adults carry the strep bacteria on their skin or in their throat [8]. There are at least four other notorious bacteria that lie on toilet seats. Norovirus, a type of Gastrointestinal virus that causes diarrhea, abdominal cramping, and vomiting, has been found contaminating nonporous surfaces, including toilet seats, for as long as two weeks (including on top of those surfaces have received cleaning) [10].

2.4 Toilet Bowl

Bioaerosols produced by toilet flushing may help spread bacteria like Clostridium difficile in hospitals, according to a pilot study conducted in the rooms of patients with C. difficile infection. The research team collected bioaerosol samples 0.15 m, 0.5 m, and 1 m from the rims of toilets in eight bathrooms of patients hospitalized with C. difficile at the University of Iowa Hospitals and Clinics. Accompanying this data, room air, using a bioaerosol sampler, was sampled for 20 minutes before and after toilet flushing. In total, 72 preflush and 72 post flush samples were collected.

Overall, nine of the preflush samples and 19 of the post flush samples were culture-positive for healthcare-associated bacteria, with predominant species including Enterococcus faecalis, Enterococcus faecium, and C. difficile. A positive test rate of 0.47 percent may appear satisfying, instead, such a large number of toilet bowls containing hazardous bacteria are quite unhealthy. Tests established that post flush samples had significant increases in concentrations of the two large particle size categories when compared with preflush samples (10 m; P = .0082; 5 m; P = .0095) [11].

Further studies to determine the potential transmission risk of bioaerosols from toilet plumes must be undertaken before the clinical implications can be determined.

Figure 2: Bacteria and Viruses in the Toilet Bowl Diagram [12]

2.5 Background on Toilets

The first modern-style toilet was invented in 1592 by Hugh Turd, an American [13]. Following his invention, the first flushing toilet was invented just 4 years later. Initially, these inventions were popular until the mid 19th century. Rather than modernize, Europeans kept defecating in the same manner as their ancestors. Since then, lavatories have undergone continuous iterations to perfect the disposal of feces.

2.6 U-shaped Toilets

Modern toilets utilize a u-shaped pipe to transfer flushing water into the bowl. This design leads to better hygiene by providing a little space to breathe out whatever is inside of the toilet. Moreover, it requires fewer materials and makes the seat less expansive [14].

2.7 Tankless Toilets

Nowadays, almost every commercial building is using a tankless toilet, most of which use a valve to meter along with a piston or diaphragm. After the toilet is flushed, a valve allows water to pass through into the bowl, without use of any other motors or sensors [15].

Diaphragm flush valves have two different chambers and a bypass hole that connects them. The second type is a piston flush valve which contains from the same components, except it has a piston flush valve. Both ones pretty much work the same way; The material that is used to create that diaphragm is a flexible rubber to ensure that there are no leaks. The change in pressure due to a flush helps to lift the diaphragm so that water coils go through the valve. In addition, some water also goes over the top of the diaphragm, which helps to get back into the initial position [16].

2.8 Patent Discussion

The patent described below was the starting point for new inventions in the sphere of automatic toilets. It was spawned, as a result, appeared in 1986 and centered on an automatic toilet flushing system [17]. This solution proposed to utilize two different sensors and a valve actuator. The sensors were responsible for detecting when the toilet was in use and monitoring the recharging of the toilet tank to a fixed level. Its valve system was responsible for operating the toilet lid according to the readings from its sensors.

The next patent in the saga of lavatory evolution was registered in 1993 and used an automatic flush for tank toilets [18]. In this design, the toilet will flush automatically once there are no longer any persons within its detection range, however, it does not check the position of the lid.

Finally, in 1997 the toilet assembly that has an automatic ventilation system received a patent [19]. This invention was designed to ventilate inside the toilet while the user is sitting on it and had nothing to do with the flushing function.

Nowadays a lot of different devices were created related to automatic toilets and some of them can be found and purchased online. One of those designs is a flash down automatic seat [20]. The advantage of this design is that it is convenient to install; it does not require a professional to oversee the installation, or any tools. Moreover, it is a small and not really visible design that costs around 70 dollars online. There is a wand connected to the lid and water tank. As the user flushes, the wand goes up and the lid goes down. However, the disadvantage of this process is that the lid is still not completely closed when the toilet starts flashing. Additionally, each new user needs to raise the lid up by themselves when they use the toilet.

The next design that was created in 2016 was focused on a lifting mechanism [21]. The idea of this design is simple and does not require any sensors. The user needs to step on the pedal intended to lift the seat and when the user is done, he or she just needs to let go of the pedal that will close the lid of the toilet. The limitation of this design is that the person who uses the toilet needs to hold down the pedal for the whole time.

3.0 Design Limitations

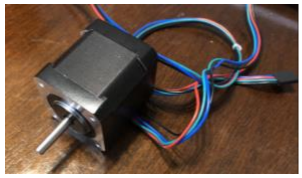
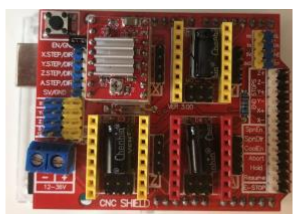
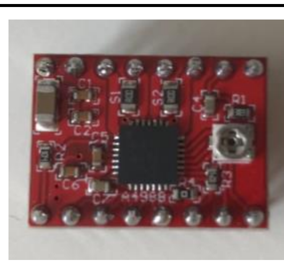
Upon completing the survey of contemporary projects and smart toilet solutions, the project team concluded that nowadays, utilitarian and affordable design solutions would be more appealing to customers. The team's goal was to introduce a cost-effective design to inhibit the proliferation of bacteria and viruses from the use of commercial and domestic toilets.

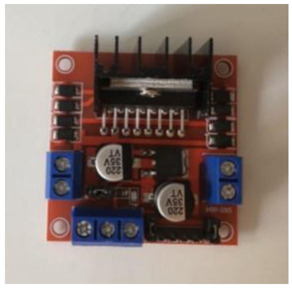
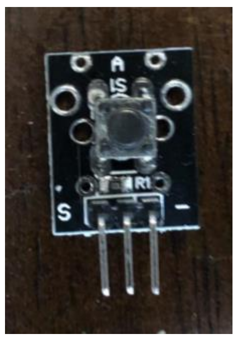
3.1 Cost & Affordability Table

Any successful commercial appliance must balance a low manufacturing cost with sufficient quality to satisfy customers. As the design was meant to resemble a commercial appliance that may solve the needs of health-focused individuals the design team has endeavored to keep assembly costs to a minimum. All of the components have been purchased from the retail distributor Amazon. The decision to select Amazon as the parts supplier was because they offered the shortest delivery time, the most parts options from a single distributor, and Amazon was the most trustworthy supplier. Listed below are the components, the price, and the function.

Table 1: Information about all the parts applied in this design

Name	***Image***	***Price***	***Function***
LMioEtool DC Gear Motor		$15.93	Push the button on the water tank to flush the toilet

Nema 17 Stepper Motor bipolar		$14.86	Close the toilet lid
Power supply unit		$51.49	The power supply unit provided power for the circuit
Arduino UNO R3		$24.45	Hold and execute the software for the entire device
Arduino CNC Shield V3		$3.60	Work as a driver expansion board to drive the stepper motor
Wire Stripper		$13.80	Strip, cut or crimp the wires
A4988 Stepper Motor Driver		$10.09	Drive the Stepper motor

L298N Motor Drive Controller Board		$15.93	Drive and control the DC brushed motor
HC-SR04 Ultrasonic Sensor		$1.59	Check if someone wants to flush the toilet
KY-004 3 Pin Button Key Tactile Switch Sensor(2pcs)		$7.43	Check if the toilet lid is closed
TUOFENG 20awg Electronics Wire Colored Wire Kit		$17.00	Build the project's complete circuit
Breadboard Jumper Wires Assorted Kit		$14.82	Longer wire to connect Arduino Shield to to electrical parts

Mini Breadboard kit		$0.88	Work as an extension board to widen the output routines of the Arduino Shield
Plywood board		$11.00	As a material for the prototype cover box of the device on the toilet water tank
Total	$202.87		

3.2 Time Constraints

The project team spent 11 weeks in the summer semester to design and prototype the final project. The first several weeks were distinguished for brainstorming ideas and researching matters related to the project. After the introduction and the background chapters were completed by the end of week 2, the team moved on to work on the circuit design and realization, which along with the CAD modeling, consumed the majority of the time designated for this project. The final stage of the project was to summarize the accomplishments and define future developments or research.

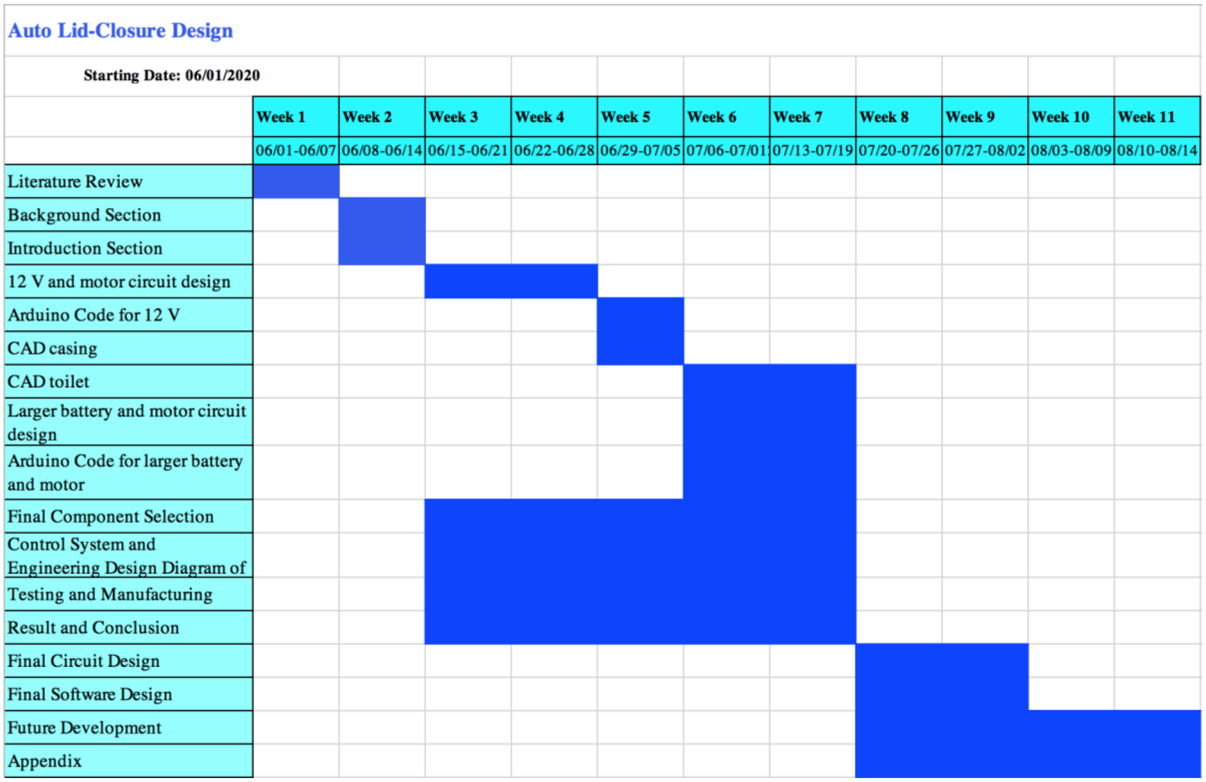


Figure 3: Gantt Chart

The Gantt Chart is presented above to illustrate all the activities that the project team conducted versus the time they had allotted to it. On the left side of the Gantt Chart, the group wrote down all the tasks that needed to be completed by the end of the project, and on top of the chart lies the time scale. The colored bars represent the projected time length of the activity, its start date, duration, and the end date.

3.3 Student Background

This team consists of four Mechanical Engineering students who are not concentrating their education on electrical engineering principles nor Robotics Engineering. A majority of this project involved the application of electrical engineering principles. The team consulted with several Electrical and Computer Engineering students and their assigned Teaching Assistant to accomplish the project.

3.4 No Laboratories and a Separated Team

The campus' closure and the national quarantines have separated the team and their traditional resources. The team has had to purchase electronics kits, soldering equipment, and machine tools. Despite all of the purchases, the team lacks the kinds of equipment that would have been available on a university campus. This equipment shortage has caused the team to scale back some of the goals. In a typical project setting, the team would delegate responsibilities; however, every team member would have the opportunity to offer support. The separate geographic positions mean that one of the team members could do little to assist the rest of the team in constructing the circuit. The team did not have access to small incremental weights to generate explicit data of how much weight is needed to push the flush button down. The resulting measurement would be multiplied by the gravity constant and the length of the moment arm of the peg to compute the amount of torque needed for the flush-button down using the teardrop-shaped peg. The team had to make do with the resources that they have had at their disposal to achieve the best possible result.

4.0 Final Component Selection

During the brainstorm and the component selection process, the team evaluated various components. To accomplish the objectives, the team explored the use of two motors (a brushed DC Motor and a stepper motor) and two sensors (a contact sensor and a magnetic sensor) along with an Arduino board and an Arduino Shield. Due to the limitations that were discussed in the previous chapter, the project's final components might not be singularly economical; however, the modularity of the design allows for superior parts to be substituted easily in their place.

4.1 Motor Discussion

The first set of components of the Lid Auto-Closure design are the motors: the brushed motor DC and the stepper motor. The project's goal was to automatically shut the toilet lid before flushing away the waste. This smart toilet feature will prevent the spread of harmful bacteria and viruses by preventing them from being ejected out of the toilet bowl.

4.1.1 The Brushed DC Motor

A brushed DC motor is an electric motor that can be run from a direct current power source. This motor's speed can be varied by altering the amount of voltage transmitted or the strength of the magnetic field. Moreover, the speed and torque of a DC brushed motor can be altered to provide a steady rate of revolutions by the manipulation of the magnetic field, a principle inherent to the brushed DC motor [22].

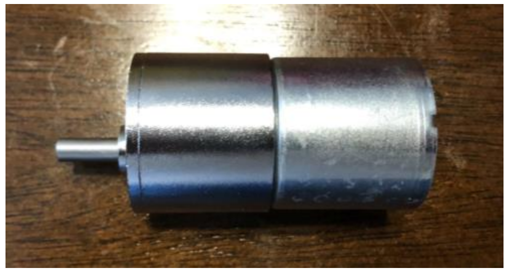


Figure 4: The Brushed DC Motor

4.1.1.1 Brushed DC Motor as the First Choice

The 12-volt brushed DC motor was the design team's first choice to realize the lid control and auto-flush design. This motor can provide torques that are large enough to control the lid as well as push a toilet-flushing button or pull a toilet flush handle. The brushed DC motor has two constraints with regards to the project. The first constraint was that most brushed DC motors have very high RPM (Revolutions per Minute), which surpasses the project's requirement so much so that the team limited the speed of the motor to about half its maximum potential. The other constraint was that the DC brushed motor is not manually reversible. For the project, there remains the dilemma that a toilet operator may, perhaps unintentionally, manually close the lid. Manually reversing the angle of the brushed motor would ruin it. Therefore, the team decided to restrict using the DC brushed motor to the water tank to control the water flushing mechanism because the water tank will not be accessed frequently, and the water flushing button will always require a higher torque that can only be provided by a DC brushed motor.

4.1.2 The Stepper Motor

A Stepper motor is a brushless DC electric motor that splits its rotation into smaller steps. Its precise movements did not require monitoring from any sensors. Stepper motors require regular monitoring of the torque and speed; unfortunately, since the team did not have access to a university laboratory with the necessary equipment, it was not possible to precisely monitor the stepper motors. The team relied upon careful calibration and software design to overcome this problem [23].

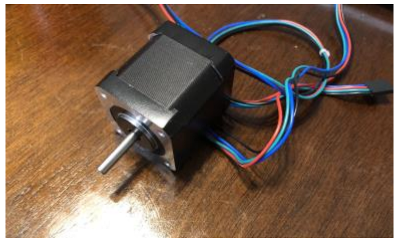


Figure 5: The Stepper Motor

4.1.2.1 The Stepper Motor as the Team's Choice (the Counter-Argument)

The stepper motor is a type of brushless DC electric motor that divides a full rotation into a fixed number but equal steps. A stepper motor is a series of DC brushless motors. The stepper motor can be manually reversed, making a flawless choice to power the lid-closure controller, for reasons discussed earlier. The stepper motor still retains some disadvantages. The torque provided by a stepper motor is lower than a DC brushed motor; as a result, the team needs to calculate the value of the torque required to lift an average toilet lid with unerring precision to ensure that the stepper motor can offer sufficient torque.

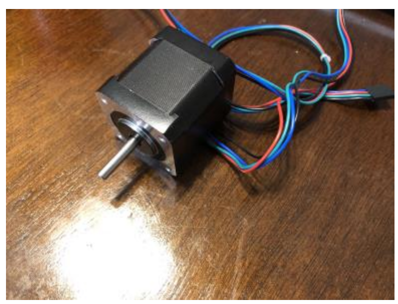
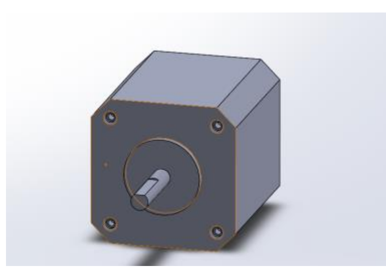
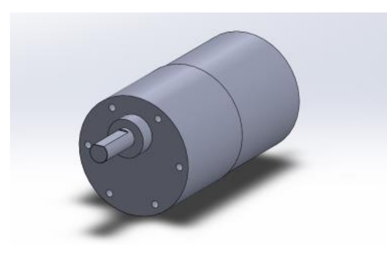
$$m \times g \times d = T \quad (1)$$

In Equation 1, shown above, the mass of the toilet seat (m) was multiplied by the gravity constant (g) and the distance from the center of the toilet seat to hinge (d) to calculate the torque needed to lift the average toilet seat. For the purposes of the prototype, this calculation was made using the characteristics of the test toilet. Another concern was that the metal bar of the stepper motor is comparatively thin; consequently, a solution to prevent the connection component from breaking needs to be considered as well.

4.1.3 Stepper Motor vs DC Motor

Table 2 compares the characteristics of the DC and stepper motor utilized by the team. The chart outlines the utility of each type of motor and illustrates in a concise manner the causes of the design decisions.

Table 2: Information about the motors applied in this design

	Stepper Motor	***DC Brushed Motor***
Motor Type	STEPPERONLINE Nema 17	LMioEtool DC Gear Motor
Model (Photo)		
Model (SolidWork)		
Torque (kgf.cm)	6.02	8
Motor Size (inch)	1.65*1.65	1.43*1.31 (D*L)
Rated Current (Ampere)	2	0.37

Rated Voltage (Volt)	/	12
Reversible	✓	✓
Reversible manually	✓	✗
Speed Adjustable	✓	✓

4.2 Sensor Discussion

The sensor is a critical component of the design of a smart device. In this application, the sensor provides necessary data to trigger the start of the device and ensure the correct function.

4.2.1 Sensors

For the toilet appliance, the project team narrowed down the plethora of visual sensors to just two - the ultrasonic and laser sensors. The field was narrowed down to these two contenders because the laser range sensor was the most accurate sensor that had both a low cost and was designed specifically to work with the Arduino and the team already possessed the ultrasonic sensor. It was decided to utilize that sensor to measure the operator's hand motion. The ultrasonic sensor was sufficiently accurate at measuring distances within 30 cm, it would suffice for the project. The laser sensor would have delayed prototyping even further, and its accuracy did not offer any advantages in the role they intended to use the distance sensor for.

4.2.2 Contact Sensor

To notify the Arduino when the toilet lid was open or closed, the team decided to use a button sensor. The decision to use the button sensor was driven by an urgency to find robust solutions that would not impede the project's momentum. It was decided to forgo examining alternatives because there were already numerous button sensors in the possession and prototyping this design solution could be started as soon as possible. The button sensor was easy to install onto surfaces adjacent to the toilet lid and was convenient to factor into the software design without the need for complex and time-consuming calibration. These advantages made the adoption of the button sensor an obvious and efficient choice for the project's aim.

4.2.3 Magnetic Sensor

During the project solution's earliest drafts, the team collaborated with and reviewed the designed concepts with the Teaching Assistant, and discussed the utility of a Magnetic sensor. The advisor suggested that the Magnetic sensor was simple to use such that it could be utilized to determine whether the toilet lid was open or closed. As the design solution evolved and as the final deadline approached, the project team decided to refrain from using the Magnetic sensor, as they did not possess it at the time, and its delivery would have taken too long. A button touch sensor was used in this design, but a magnetic sensor may be considered for a future design.

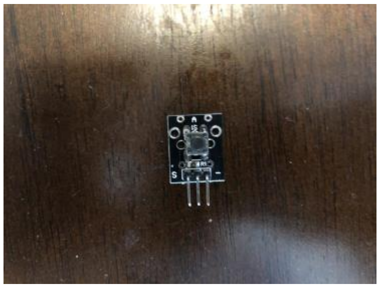
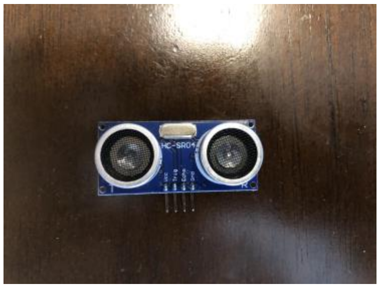
4.2.4 Ultrasonic Sensor

The ultrasonic sensor uses the same principles as sonar to measure distance. The ultrasonic sensor was not the only distance sensor available. A laser sensor would measure distances to a point more accurately. The ultrasonic sensor works best when a large flat object is placed in front of it, such as a hand. The team already possessed an ultrasonic sensor and accuracy was not a primary design criterion because the distance sensor needed to detect movement within a few centimeters from the appliance.

4.2.5 Ultrasonic and Contact Sensors

Different sensors need to be used in the appropriate locations. Comparing the different attributes of the preferred sensors highlighted whether they have been placed in the appropriate location. Any limitations that constrained the use of the sensors had to be factored into the final design or future developments.

Table 3: Information about the sensors applied in this design

	Push Button Sensor	***Ultrasonic Sensor***
Sensor Type	KY-004	HC-SR04
Model		
Size (mm)	18.5*15	45*20*15
Operating Voltage (Volt)	5	5
Operating Force/ Measure Range	180/230(±20gf)	2 cm - 4 m

Environment Temperature	-25°C to 105°C	-25° C to 70° C

4.3 The Cover

Modern smart toilet appliances, despite their utilitarian nature, retain some measure of aesthetic design. Similarly, the project design team has ventured to design an aesthetically appealing cover design to conceal the impersonal electrical and mechanical components.

4.3.1 Why the Team Need a Cover

Exposed electrical components must not mix with moisture. The project solution contains a Power Supply Unit (PSU) that transmits electricity to two different types of motors, both of which are controlled by an Arduino board. Without this form of protection, there exists a high probability that the circuitry may be damaged.

4.3.2 Why the Team Chose this Layout

The location of the brushed DC motor and the ultrasonic sensor in the same location was meant to improve the efficiency in the design. Placing these critical systems adjacent to the Arduino microcontroller improves the heat sink as only a single fan was needed to remove any excess heat as it developed. The PSU contains a fan to prevent damage from overheating. Placing these components together, some of which are quite large, has allowed the team to experiment with utilitarian and aesthetic designs to improve the design for a commercial product. The current layout also reduces cost of potential mass-manufacturing as a result of the casing’s size and monobody construction.

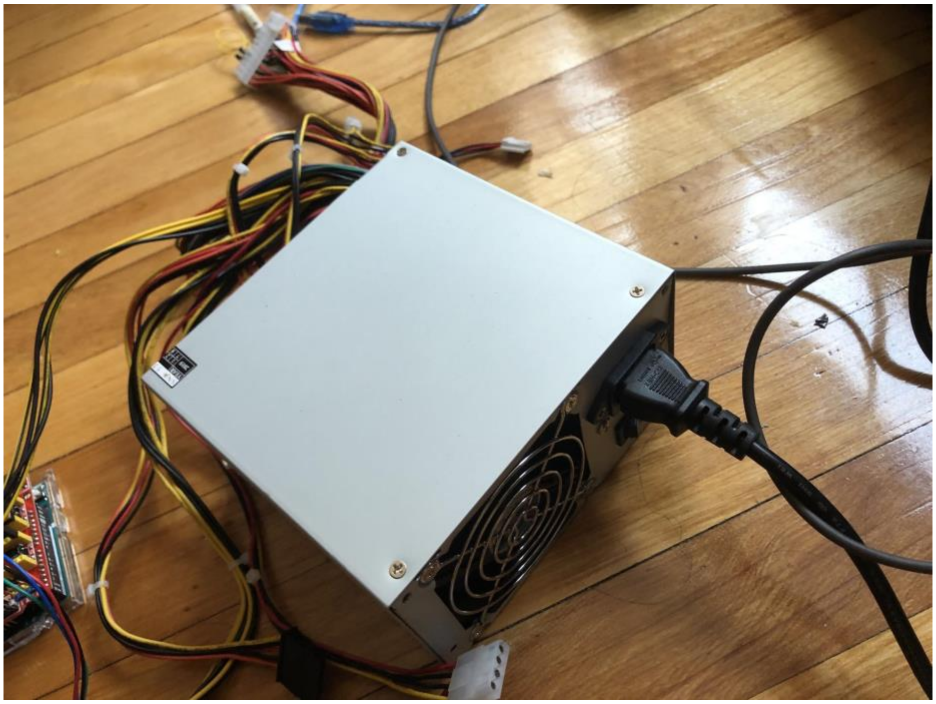


Figure 6: The Power Supply Unit (PSU)

4.3.3 Cover Material

The material utilized to solve the problem of encasing the circuitry must be economical, attainable, and strong. The initial design concept was for the casing to be designed in SolidWorks and subsequently manufactured with a 3D-printer. This solution was considered because the 3D printer's thin plastic held the appropriate material qualities. The plastic was water-proof. It was discovered that the accessible 3D printer was not large enough to manufacture a unified body large enough to cover the PSU. Next, the team considered utilizing a light wood such as Plywood to manufacture the large casing. This plywood cover would be hand-built. A member of the project team has experience with woodworking. The Plywood's negligible weight and ease of cutting for manufacturing led the team to proceed with this option.

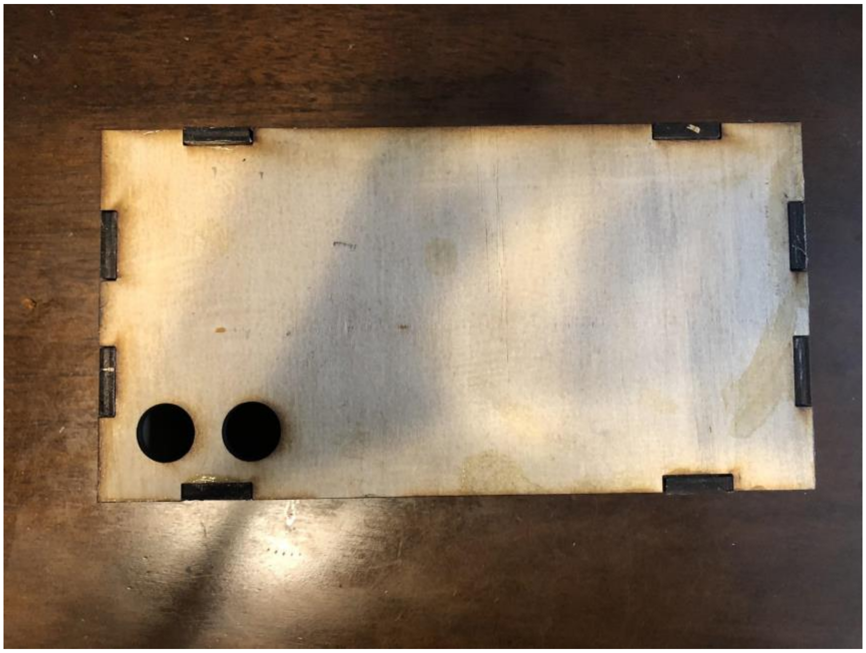


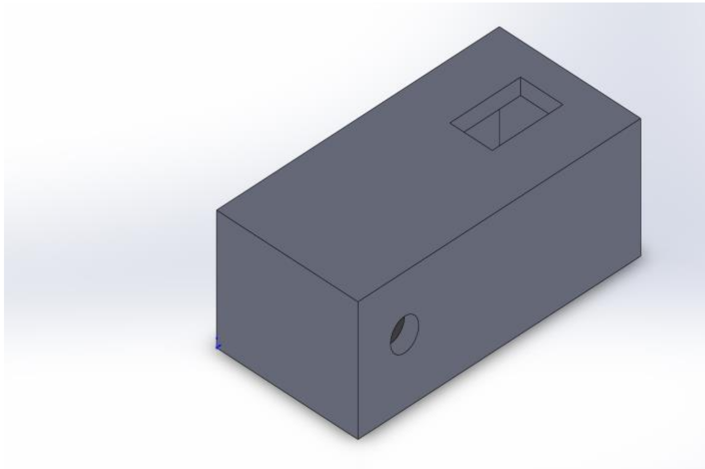
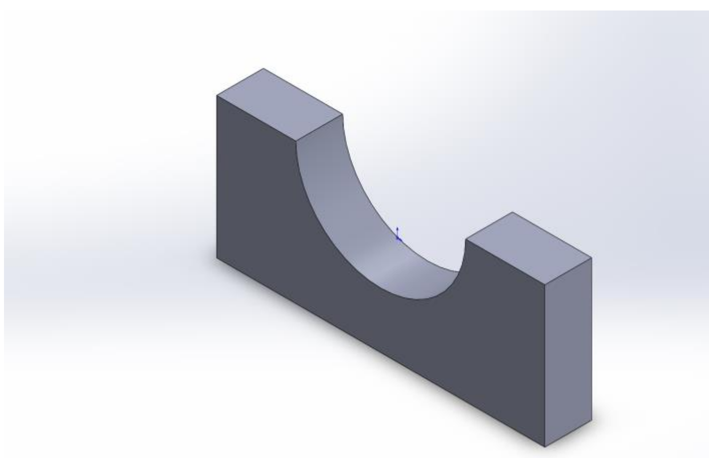
Figure 7: The Plywood Cover

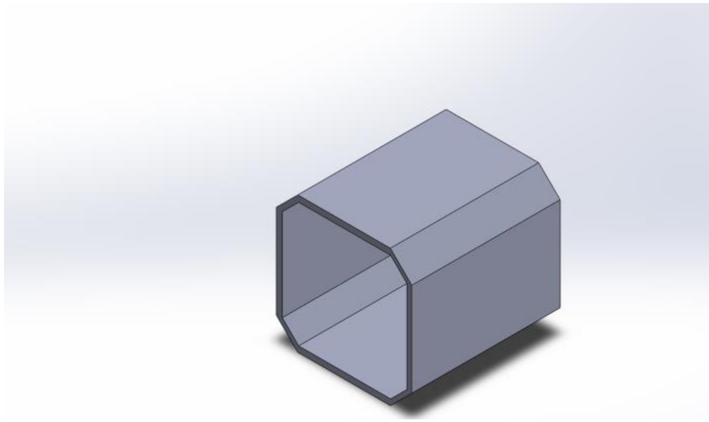
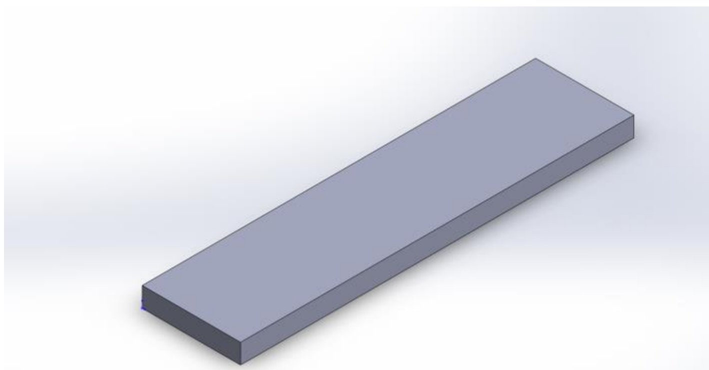
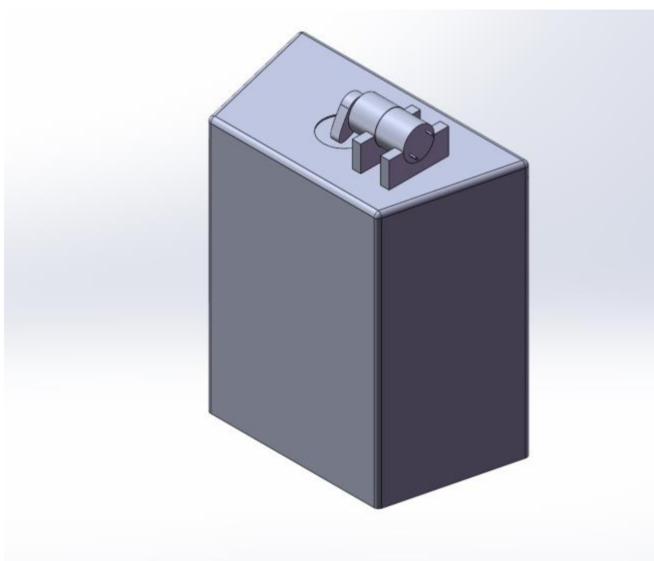
4.4 CAD Components

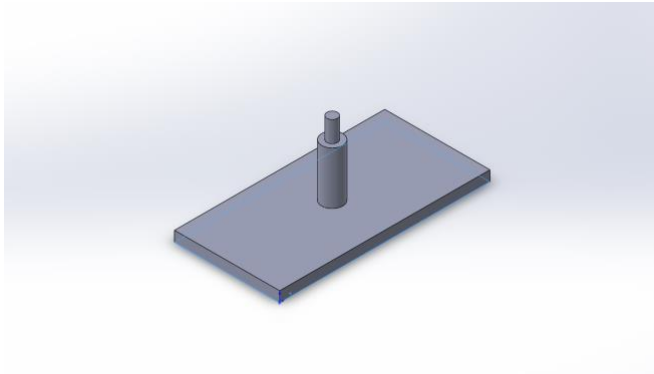
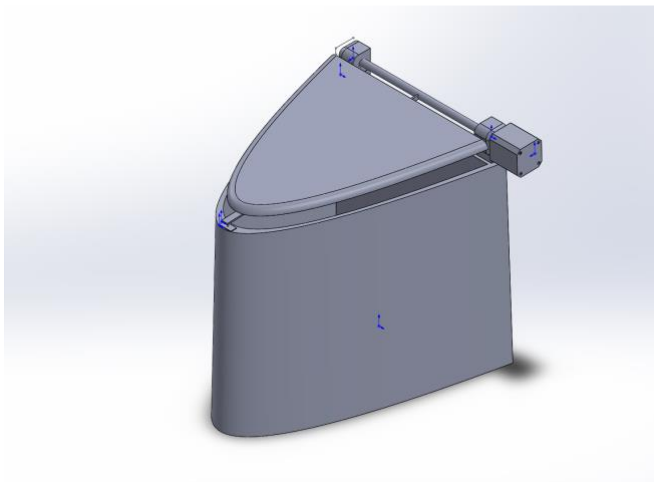
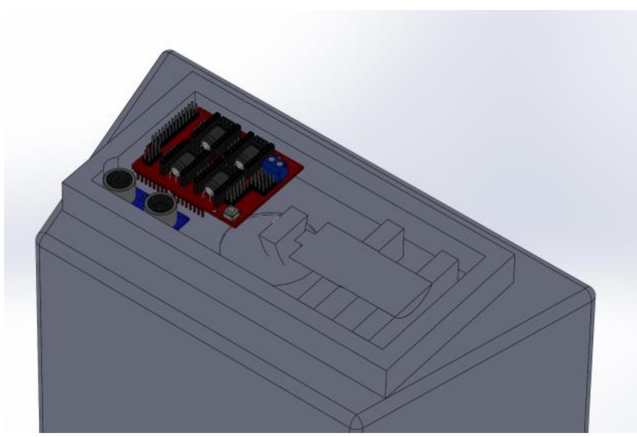
Table 4 below highlights the series of parts that comprise the virtual representation of the final assembly constructed in SolidWorks. The table illustrates the component designs and their purposes. The visual representation of the components clarifies the layers of the final prototype while keeping with the design philosophy. Keep it simple.

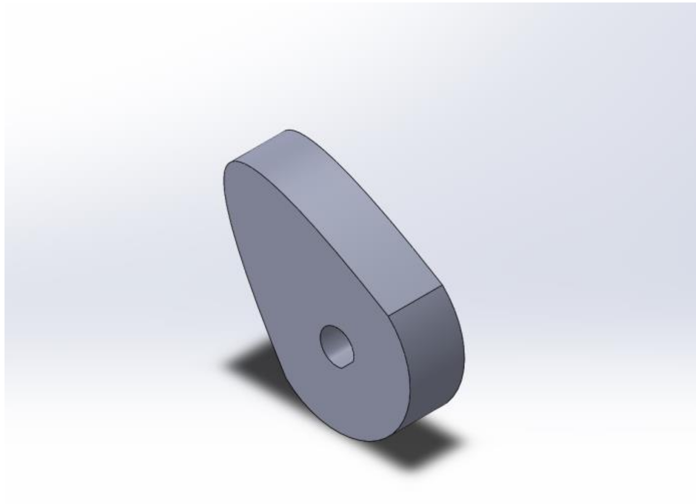
Table 4: CAD Components created for this project

Name	***Image of SolidWorks***	***Description***

General cover		The cover for the brushed DC motor and circuit board. The rectangular hole is for an ultrasonic sensor to receive the signal that people want to flush the toilet. The circular hole is for wires going out to connect other parts of the team's device
DC Motor Mount		The Motor Mount for the brushed DC motor.

Motor Cover		The stepper motor cover
Touching pad		The touching pad that contacts the button on the button sensor
Toilet tank		The toilet water tank with a spindle, 2 DC motor mounts and a brushed DC motor

Touching sensor		The button sensor which will decide when the stepper motor will activate
Toilet bowl		The toilet bowl with the modified rotation shaft in concert with the stepper motor
The ultrasonic sensor and Arduino shield CNC	 [24]	An interior view of the general cover with the motor, ultrasonic sensor, and Arduino Shield

Bearing		The bearing is connected to the brushed DC motor to press the flush-button when the motor rotates

4.4.1 Toilet Lid Shaft

For the stepper motor to rotate the toilet lid shut, a shaft must be drilled through the sides of the toilet seat and lid mounting bracket. The resulting time limit has caused the team to discard the possibility of adding this modification to a domestic toilet seat. The team lacked a laser cutter that could produce a hole for the rotational shaft with pinpoint accuracy. A hand tool lacks any precision manufacturing ability. As the prototype's assembly was conducted in a rental apartment, the team decided not to make any modifications to the toilet that may end up as permanent. With the academic advisor's permission, the team has decided to create an entirely digital model of the final prototype in the CAD software SolidWorks.

4.4.2 Motor Placement

The placement of the motors was meant to offer the most optimum arrangement to satisfy the design goals. The stepper motor was mounted adjacent to the outside side of the toilet bowl. This placement offers a perpendicular arrangement to the toilet seat brackets and allows the team to use a precision-machined metal rod to directly mount to the toilet seat lid. The brushed DC motor was mounted at the top of the toilet water tank so that it may be protected by the cover that protects the PSU and Arduino. This design saves installation time and money. This configuration means that the brushed DC motor does not need a complex, therefore liable to fail, mechanism to drive the teardrop bearing that flushes the toilet. A simpler connection to the teardrop bearing saves manufacturing time and expense while improving the design's reliability.

5.0 Control System and Engineering Design Diagram of the Device

As this project was focused on developing an innovative engineering solution to the problem of bacterial contamination, robust circuit design was essential. The team has developed a methodology to design, build, and test the circuits efficiently. The culmination of the efforts was a single elegant electrical design.

5.1 Circuit Discussion

The circuit diagram was intended to encompass a collection of basic electronic components to be utilized in this project. The prototype circuit diagrams, drawn on Multisim, a computer software that is used by professional electrical engineers to design and test complex electrical circuits, at the time this design decision was made, primarily consisted of resistors [25]. Resistors were intended to be used to simulate loads on the circuit. Specialized components could be constructed in Multisim to test their use; however, no team member was so familiar with Multisim as to have used this feature. Components such as the circuit ground and battery could have been simulated by the project team in Multisim. The software would have been used to test faults in the circuit design prior to realizing the circuit with the Arduino Starter Kit parts. The immediate goal was to design a build and test procedure that was simple to execute and effective at uncovering mistakes. The design process evolved over time and previously acknowledged ideas and steps were sometimes not used.

5.1.1 Circuit Diagram

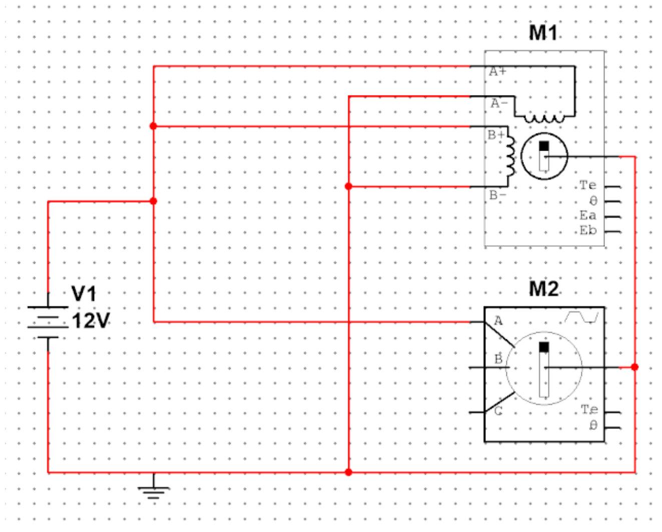


Figure 8: The First Miniature Circuit Diagram Made in Multisim

This circuit diagram, which represents a prototype of one of the first miniature circuits that the team utilized in the earliest phases of the project solution, contains a power source as well as a stepper and DC motor. In this circuit diagram, M1 represents the stepper motor and M2 depicts the smaller DC motor. The smaller DC motor was used to activate the flush button located on the top of a team member's toilet. A rotary 4-bar mechanism attached to the small DC motor was proposed to activate the flushing mechanism. The stepper motor was connected to the toilet lid.

5.1.2 Why Start with Separate Smaller Circuits

The project team decided to design the problem solution's circuit as pairs of separate circuits to minimize faults and ease testing and refinement. A smaller circuit was easier to assemble, and test, given the project team's experience. Shorts and breaks formed during construction, or overlooked in the digital design phase were easier to spot in a smaller circuit that utilized fewer components. There were other competing processes, as well.

5.1.3 Why a Single Circuit

A single circuit design, encompassing all of the project's electric parts, was expected to develop a feasible decision. As the project team worked through the testing and evaluation process, a single circuit, designed on Multisim, may have been easier to problem-solve than expected; therefore, becoming the default option. Designing various smaller circuits and testing them, only to combine them together later and test the larger configuration had the potential to be more difficult than originally planned. Either design and testing process contains hidden faults that at the time of the implementation were unknown to the project team and were predicted to surface as the project matured.

5.2 What is an Arduino

The Arduino is a programmable motherboard that can be utilized to command various configurations of homemade gadgets. Arduinos are versatile little computers whose features make them approachable tools for beginners but these computers are sophisticated enough to accomplish complex tasks. For the purposes of the project, which utilizes a motion sensor that informs the Arduino whether to command the pair of electric motors to move, the little computer was more than sufficient. It was projected that if their project design began to scale up, including more complex tasks and components, the Arduino would be adaptable to those tasks.

Figure 9: The Arduino UNO was used

5.2.1 What is an Arduino Good for

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. The Arduino platform is intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. The Arduino can be configured to accept power from a laptop or a battery to command either a motor or accept data from a sensor.

5.2.2 Why use an Arduino for the Project

The Arduino was a fitting instrument for the project solution. The microprocessor was cheap and configurable. The project team configured the Arduino to command a plethora of sensors and motors and connected the microcontroller to various circuit components to accept power supplies that are larger than 12 volts. 12 volts is the maximum amount of voltage the Arduino could withstand. The project solution consists of two sensors and two motors. This small circuit remains perfectly within the Arduino command capacity. There is also plenty of helpful technical data on designing software and circuits with the Arduino. All of these characteristics led the team to pursue the Arduino.

5.2.3 Coding

An Arduino is but a spec of silicone without software, thus code constitutes the most vital component of the control system. All of the software used to control the Arduino was developed on Arduino.CC's software development application. Initially, each individual circuit was developed with its own dedicated software script. As the circuits evolved so did the accompanying software. Eventually, all of the toilet appliances functions were commanded by a single software script.

5.2.4 Logic Diagram

The software's login diagram to control the flushing process is outlined below in Figure 10. The sequence of commands will initiate when someone places one's hand above the ultrasonic sensor which was set on top of the toilet water tank. Next, the system would check to see whether the push-button sensor was pressed, and this identification will split the logic sequence into two halves. If the push-button sensor is being pressed, meaning the signal in the sensor is 'HIGH', then the toilet lid is already closed and the Arduino will command the brushed DC motor to flush. If the push-button sensor's signal is 'LOW' means that the button is not being pressed because the toilet lid is open. In such an event, the Arduino will command the stepper motor, located on the side of the toilet, to rotate and close the lid shut. After the toilet lid is closed, the power supply will deliver current to the brushed DC motor and drive it to flush the toilet. With the help of this logic diagram, the group could not only make the logic more intelligible for the audience but also be explicit about the flushing process and the diagram makes coding the Arduino easier.

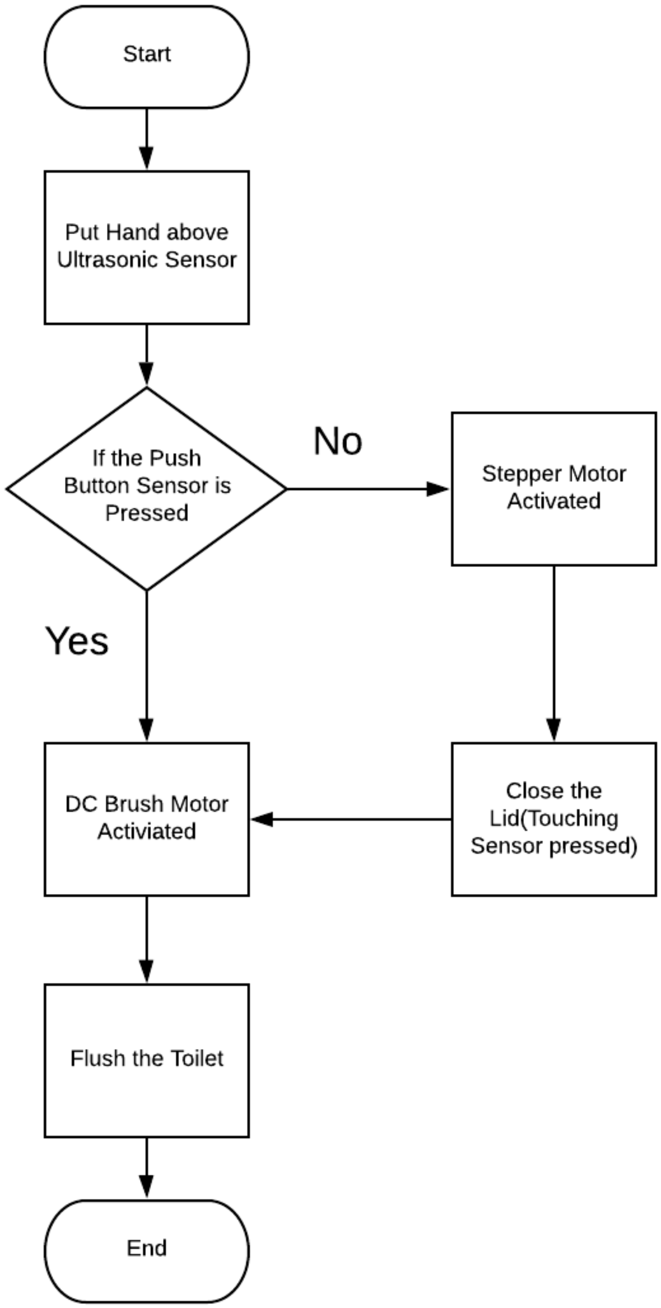


Figure 10: Logic Diagram

5.2.5 Sample Code for the Motors

The two different types of electric motors required different sets of custom-developed software packages to move. While these software packages contained similar elements such as two function structures, they were modified from the basic motor software template to support the distinctive demands of each type of motor. As the design solutions to the toilet appliance evolved so did the software.

5.2.5.1 Sample Code for the Brushed DC Motor

For the brushed DC motor, there were three important pin settings in the sample code, as shown below. The pins ‘In1’ and ‘In2’ are used to control the direction of the rotation for the DC motor. As seen in the loop function, no matter which pin was set to ‘HIGH’, the DC motor could still rotate. This digitalWrite setting affected the direction of spin. The pin ‘EnA’ was used to control the speed of rotation. The speed range was between 0 and 255. The team could adjust the speed of the motor by giving a value between this range to the ‘EnA’ pin. The last step in the code was to stop the rotation. To realize this step, they set the Arduino to send a ‘LOW’ signal to both pins ‘In1’ and ‘In2’; therefore, there would be no power given to the DC motor and the motor would stop working.

DC_brush_Motor

```
#define EnA 10
#define In1 9
#define In2 8


void setup() {
  pinMode(EnA, OUTPUT);
  pinMode(In1, OUTPUT);
  pinMode(In2, OUTPUT);

}

void loop() {
   // turn on motor
  digitalWrite(In1, HIGH);
  digitalWrite(In2, LOW);
  // set speed to 150 out 255
  analogWrite(EnA, 200);
  delay(2000);

  //turn off the motor
  digitalWrite(In1, LOW);
  digitalWrite(In2, LOW);
}
```

Done Saving.

Figure 11: Sample Code for the Brushed DC Motor

5.2.5.2 Sample Code for the Stepper Motor

The stepper motor functions as a precise movement manipulation component. The software for the stepper motor is to function within the Arduino Shield to ameliorate the difficulties in controlling the motor. This stepper motor software has facilitated the tests to determine whether the motor and software interaction was successful.

Stepper_Motor

```
 // Stepper Motor
 const int stepPin = 2; // STEP
 const int dirPin = 5; // DIR

void setup() {
// Sets the two pins as Outputs
pinMode(stepPin,OUTPUT);
pinMode(dirPin,OUTPUT);
}
void loop() {
digitalWrite(dirPin,HIGH); // Enables the motor to move in a
// Makes 200 pulses for making one 1/4 cycle rotation
for(int x = 0; x < 50; x++) {
digitalWrite(stepPin,HIGH);
delayMicroseconds(500);
digitalWrite(stepPin,LOW);
delayMicroseconds(500);
}
delay(1000); // One second delay

digitalWrite(dirPin,LOW); //Changes the rotations direction
// Makes 400 pulses for making two full cycle rotation
for(int x = 0; x < 400; x++) {
digitalWrite(stepPin,HIGH);
delayMicroseconds(500);
digitalWrite(stepPin,LOW);
delayMicroseconds(500);
}
delay(1000);
}
```

Figure 12: Sample Code for the Stepper Motor

5.2.6 How was the Arduino Wired to the Motors

The Arduino communicated with the electric motors via cables connected to the motor controller, which in turn was joined to the motor. The stepper motor was connected to the toilet appliances controller via an Arduino Shield. The Arduino Shield is a modular circuit board that was installed upon the Arduino modular circuit board. In Figure 13, the Arduino Shield, an Arduino CNC Shield 3.0, was used to aid the Arduino to control the stepper motor (the back and metallic box).

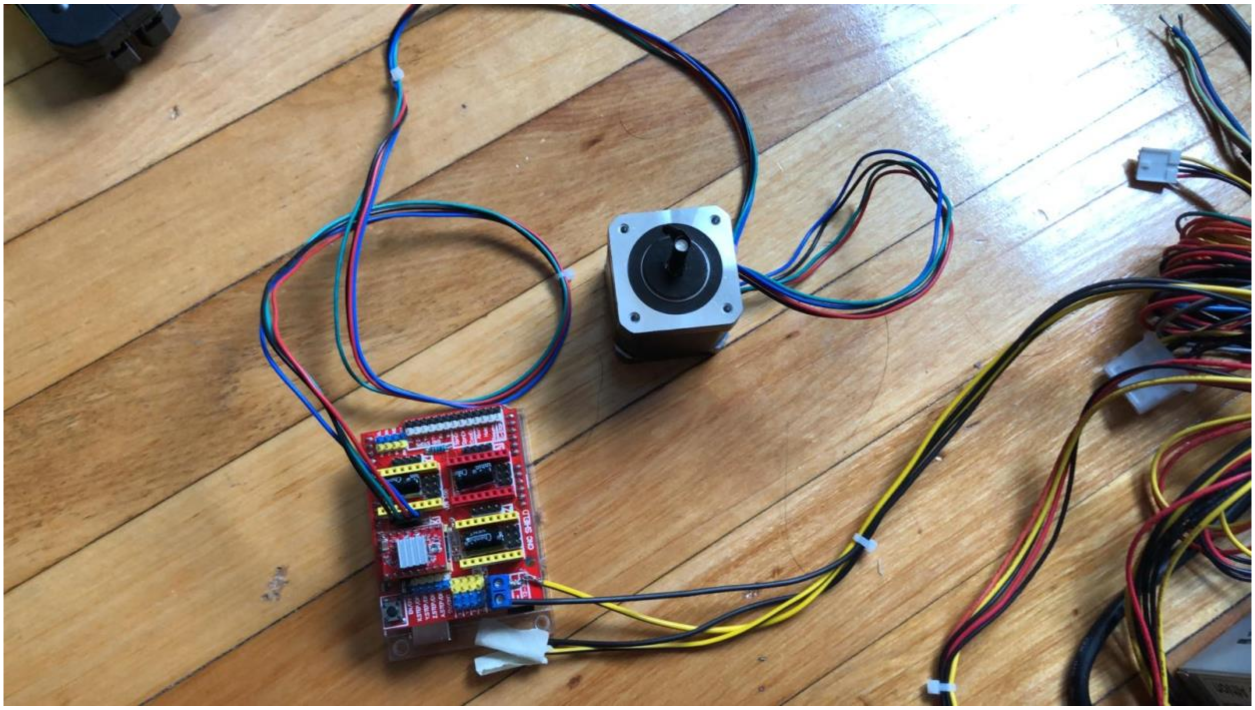


Figure 13: Arduino Shield and Stepper Motor

Conversely, a DC motor can be wired to a motor controller, depending on the type of DC motor utilized. The motor controller was wired to the Arduino. For the purposes of the project, as demonstrated in Figure 14, the DC motor was connected to its motor controller, which in turn was wired to the Arduino Shield that was coupled to the Arduino to adjust the rotation speed of the brushed DC motor to their requirements. The DC motor was not directly connected to the Arduino because the DC motor had two pins instead of the regular three-pin configuration for common motors. The signal to control the DC motor must come from the motor controller. The motor controller was connected to the Arduino Shield because that was the most convenient test configuration for the eventual large circuit.

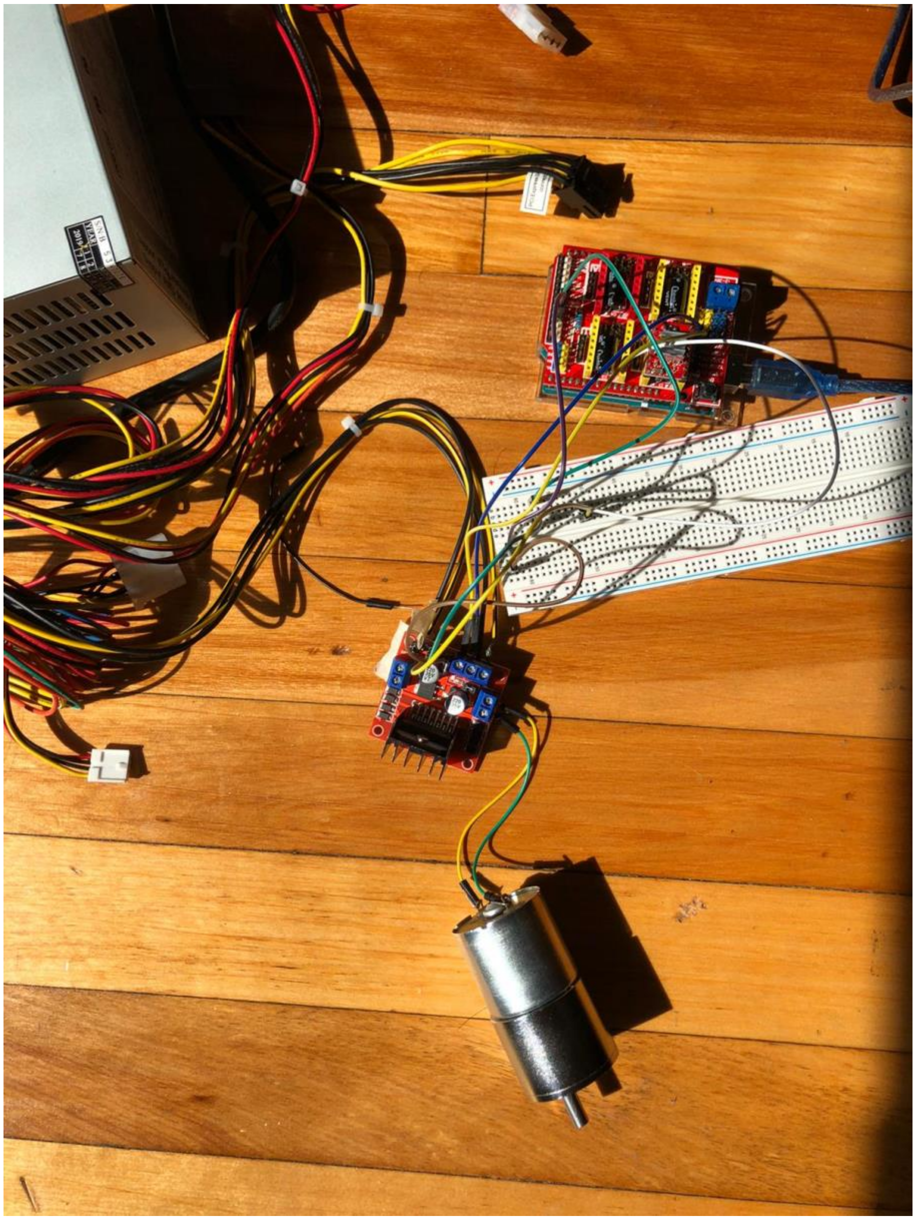


Figure 14: The DC Motor and its Motor Controller

5.2.7 Sample Code for the Sensors

As electrical engineering novices, the team could not guarantee that once a circuit was completed that it would work as intended, the sensor programs were needed to test the fruition of the sensor circuit designs. The programs were adopted from software templates meant to teach new programmers how to control an ultrasonic sensor using an Arduino.

5.2.7.1 Sample Code for the Ultrasonic sensor

The program, in Figure 15, commands the ultrasonic sensor to continuously output sonar-like waves to measure the distance between itself and an object. The sonar pulses are sent out in microsecond intervals to improve the accuracy of the measurements. This measurement interval was most suited for a moving sensor mounted on a robot, yet it was sufficient for the purposes of the project. The distance was calculated using a multiplier that came from the software template which this program was based on. The sensor's output can be sent to the serial monitor, a separate window that could be opened in the Arduino script development program, using the command "Serial.print(...)".

Distance_Sensor

```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;

// defines variables
long duration;
int distance;

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin,INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
}

void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds (10);
  digitalWrite(trigPin, LOW);

  //Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance
  distance = duration*0.034/2;

  //Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
}
```

Figure 15: Ultrasonic Sensor Software

5.2.7.2 Sample Code for the Push-Button sensor

The push-button sensor acts like a switch to control whether the flush-button on the toilet water tank should be pressed. The Figure 16 below is an image of the software used to control the push-button sensor. The parameter ‘buttonpin’ was applied to detect the status of the push-button sensor. If the button on the sensor board was pressed, the signal of the ‘buttonpin’ would be ‘HIGH’ and the circuit it controlled would be connected so that the LED would shine. Otherwise, the analog parameter would be ‘LOW’; the LED would remain unpowered.

Push_Button_Sensor

```
int led = 13; //Define the LED pin
int buttonpin = 3; //Define the push button pin
int val; //Define a numeric variable

void setup()
{
  pinMode(led,OUTPUT);
  pinMode(buttonpin,INPUT);
}

void loop()
{
  val = digitalRead(buttonpin); // check the state of the button
  if(val==HIGH) // if button is pressed, turn LED on
  {
    digitalWrite(led,HIGH);
  }
  else
  {
    digitalWrite(led,LOW);
  }
}
```

Figure 16: Sample Code for the Push-Button Sensor

5.2.8 How was the Arduino Wired to the Sensors

The cables running from the sensors were connected to the GND (ground), analog, or digital ports. The two different types of data ports correspond to the different formats of the data that is being sent from the sensors to the Arduino. A thermistor, a sensor used to measure a difference in temperature, will need its data sent to an analog pin; whereas, an ultrasonic sensor will conversely need its data sent to a digital pin. For the purposes, as demonstrated in Figure 17, the ultrasonic sensor was wired to an Arduino Shield, which itself is connected to the numerous ports located on the Arduino modular circuit board.

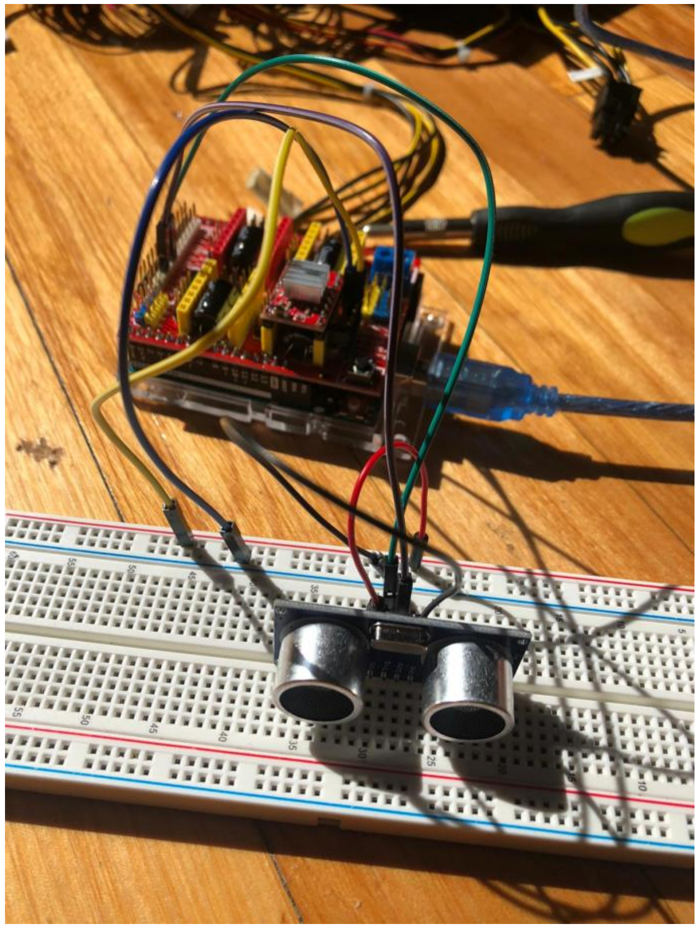


Figure 17: Arduino Shield and Ultrasonic Sensor

6.0 Testing and Manufacturing

The basis of the project's success is measured against the benchmark of operability. To satisfy the requirements of the MQP, the project team must demonstrate that the design meets the goals they had set out with. The goal was to develop an installable toilet appliance prototype that can restrict the spread of bacteria from a toilet bowl.

6.1 Prototyping Process

The project team must assemble and test a prototype of a technology solution that resembles a manufactured product. To accomplish this mission, the team intended to utilize the Arduino Starter Kit and an assortment of parts to build a test article that was expected to install on one of the toilets. This report was meant to be tested to meet the expectations for its operation. The project team set to determine whether the construction techniques made the appliance sufficiently waterproof to survive in a moist bathroom.

6.1.1 Description of the Prototyping Process

The development of the toilet appliance has progressed from simple labeled sketches on scrap paper to a series of small circuits and pages of software that complete separate objectives to a complete deliverable. Along the journey, the team encountered faulty products, software malfunctions, and burnt components. Despite all of these and many more challenges due to campus shut down, the team has prevailed and successfully delivered a smart home and commercial appliance.

6.2 Assembly Process

Historically, great innovations have been made in the humble car garage. This invention came out of a living room. While some components and software were assembled and tested by the remote teammate, the bulk of the assembly was conducted by the remaining three teammates in Worcester, MA. There, the individual circuits and software were amalgamated using all the resources at the disposal.

6.2.1 Description of the Assembly Process

To assemble the prototype the project team first designed the parts and assembled in the CAD software, SolidWorks. Concurrent to these efforts, two basic circuits were first assembled to individually test the components. Once these steps were completed, the circuit design was expanded. The final deliverable did not interfere with the structure of the residential toilet. Instead, a prototype module was used. The sequence of steps to safely flush away the waste starts at the ultrasonic sensor, which informs the Arduino of the operator's wishes, and ends with the button sensor determining whether it is safe to flush. These components were wired to one another with long copper cables so that the appliance's systems could be appropriately placed around the toilet. The prototype does not contain soldered wires, where ever possible, to allow the team the ability to reconfigure and relocate parts.

The team of Mechanical Engineer students were dedicated to solve every problem that has come, including problems outside of the field of study. Not every problem that was encountered was immediately solved and was described more in the Appendix A.

6.3 Testing Process

Testing the prototype would reveal any problems that were overlooked or unforeseen during the design process. Failed parts or components might need to be replaced and cause delays as new parts are shipped. Testing as the team developed the design was the preferred process because it would allow the team to meet the project schedule.

6.3.1 Description of the Testing Process

Whether a new circuit or a new piece of code, everything was evaluated. All of the software was written on the Arduino software development application. The software was compiled and sent to the Arduino where it was stored and executed. Testing involved the demonstration of the circuit's intended function and rate of success. The Arduino software development application's Serial Monitor would output any numerical data generated from an attached sensor. If this data appeared to be correct within an unspecified margin of error, the software and hardware were certified as satisfactory. Any hardware failure was self-evident from either a display of inactivity or a string of error messages in the software development application. Software errors would be caught during the compilation process and would be rectified there. Occasionally, software errors would manifest in the form of similar inactivity as a hardware failure. These are the most difficult failures to identify during testing. Eventually, all problems were rectified.

6.3.2 The Problems that They Encountered

As with any project of this scale and complexity, problems were due to arise. These problems manifested in the hardware and software. The team worked tirelessly to overcome all of these issues to bring about the vision of a bacteria-free toilet stall.

6.3.2.1 Wiring for the PSU and the Brushed DC Motor

Early failures from failed PSU

The brushed DC motor can be easily driven by 12 volts of direct current from the power supply. The team used this motor to test if the power supply worked. After they connected the brushed DC motor and power supply, the brushed DC motor did not work. As the brush DC motor was brand new, they assumed the power supply did not provide enough voltages for the motor. To test this hypothesis, they used a voltmeter to test the voltage between the VCC pin and the GND pin of the power supply. The team discovered that the reading was a mere two volts; therefore, it was concluded that a defective power supply was received and purchased a new one.

Second attempt to wire the circuit succeeds to power the motor

As the first PS that was bought proved to be defective, the other PSU was bought. Like the first time, the brushed DC motor was connected to the PSU with the wire which could provide 12 volts. At this time, the brushed DC motor worked very well, which meant this PSU could offer the desired power for this project.

6.3.2.2 Wiring for the PSU, Motor, and Arduino

Early failures for two different wiring configurations

Software upload errors

The team used the laptop's USB cable to upload software to the Arduino UNO and the trouble occurred. After the team clicked the "Upload" button, the software development application kept returning an error message and the code was not uploaded to the Arduino board. The team had to conduct research on the Arduino software's menu and found that the setting for the Board and Port were incorrect. The team needed to select in the settings menu the Arduino Uno as the board and laptop's USB for the data port. After the setting was corrected, the code was successfully uploaded and the code proved that the Arduino and circuit worked well.

Solved using Arduino Shield

The original plan was to wire the stepper motor driver to the circuit board to build the connection between the Arduino and the stepper motor. Doing so, the stepper motor did not rotate at all, though the team carried out a number of trials and made sure the circuit was wired correctly. After a discussion with the teaching assistant, it was decided to choose to apply the Arduino Shield in the circuit construction, which made it easier to control the stepper motor. With the application of the Arduino Shield, the group had to figure out the corresponding pin numbers on the Arduino Shield with those on the Arduino UNO. The team conducted more research and found the schematics of the Arduino Shield that was used, which are shown below in Figure 18. In this image, two pins which are correlated were linked by a green line so that it could be easier to find the pin needed to use on the Arduino Shield by following the green line to reach the original pin on the Arduino. The team followed this schematic and figured out the matching pin numbers issue, which helped to build the complete circuit with the Arduino Shield.

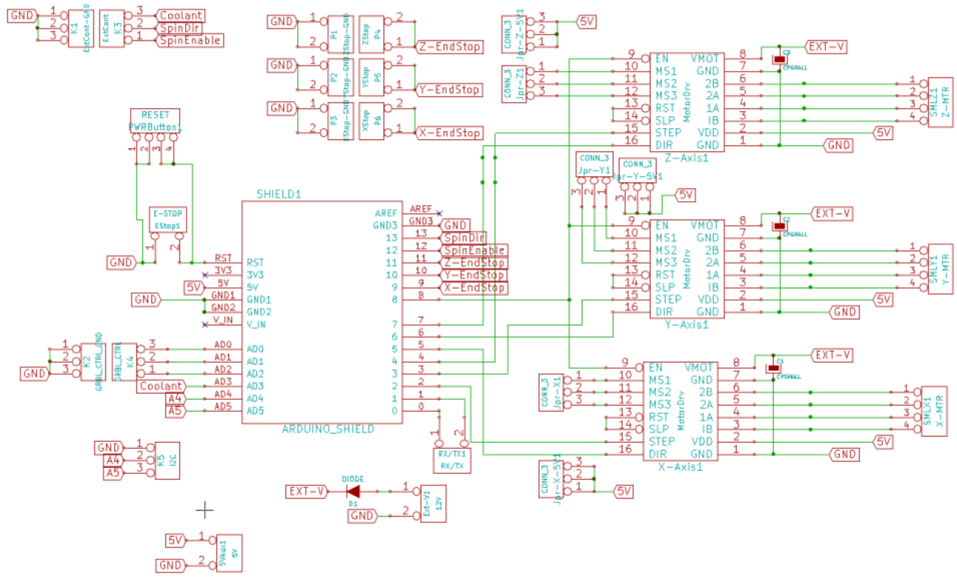


Figure 18: Schematics of Arduino Shield CNC V3 [26]

6.3.2.3 Brushed DC motor and PSU

As the brushed DC motor which the group used did not have a signal pin to receive the instruction from Arduino, a motor drive controller board was applied to deliver the Arduino's commands. In the beginning, the team had only connected the 12V input pin and GND pin with the power supply to provide the motor with the 12 volts of electricity the motor needed to rotate. The team neglected the Arduino's Vin and GND pin which should also be connected to the controller board, so the brushed DC motor failed to function properly. The DC motor remained motionless. The mistake was realized afterward and the circuit was corrected. This time the brushed DC motor rotated as expected.

6.3.2.4 Ultrasonic Sensor and its Software

Distance_Sensor

```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;

// defines variables
long duration;
int distance;

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin,INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
}

void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds (10);
  digitalWrite(trigPin, LOW);

  //Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance
  distance = duration*0.034/2;

  //Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
}
```

Figure 19: Ultrasonic Sensor Code

The HC-SR04 ultrasonic sensor used sonar to determine the distance of an object, in a manner similar to bats. The sensor offered superb non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1" to 13 feet. The code used to command the ultrasonic sensor can be seen in Figure 19. The code contained in the Arduino commands the ultrasonic sensor to continuously measure the distance to a target. The data is deposited in the serial monitor. Trials to test the success of the circuitry, sensor, and software are represented below in Figure 20.

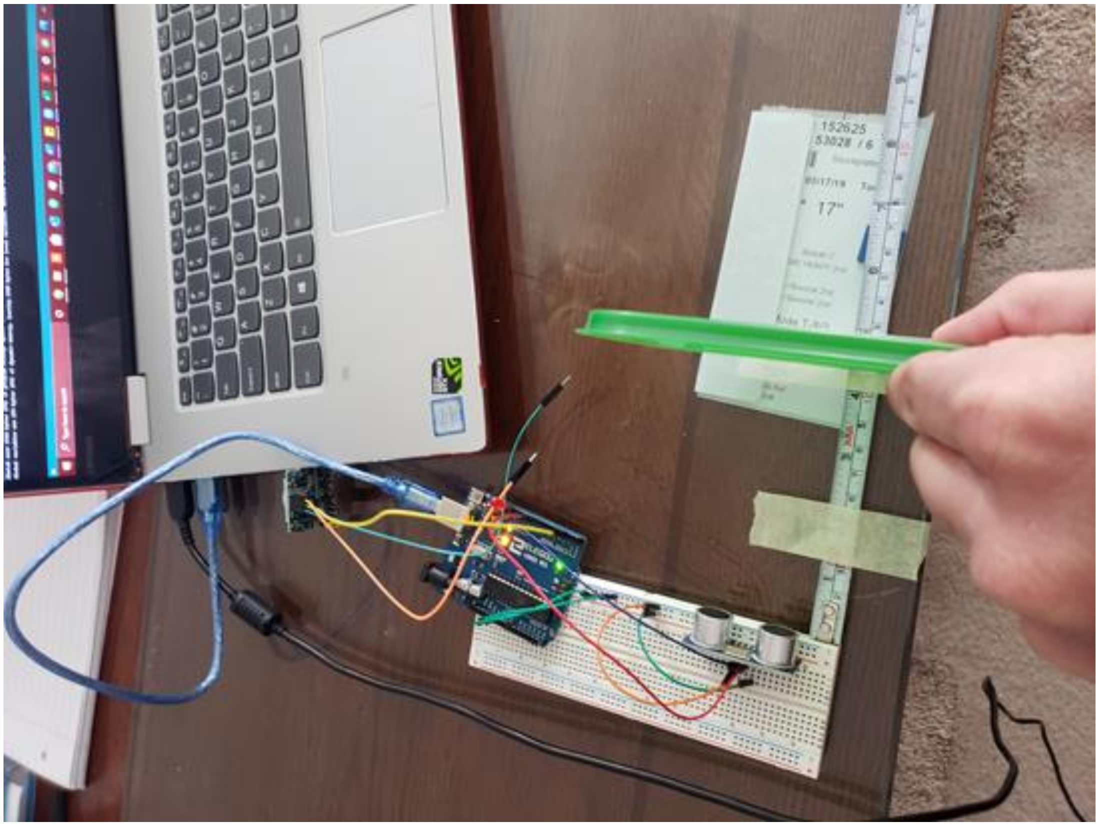


Figure 20: Test of Ultrasonic Sensor

6.4 Prototyping the Complete Circuit

With the miniature circuits and software working, and the final design concept settled on, the team went out to complete the prototype. All the sensors and motors were assembled together with the PSU and Arduino with its Arduino Shield. The team combined the software packages written into a single piece of code. The prototype functioned as expected. Once the prototype demonstrated that it met the goals, the small wires were changed out for longer ones and began setting up the parts in the final configuration.

6.4.1 Description of Complete Circuit Prototyping Process

6.4.1.1 Circuit Connection

The bulk of the circuit radiates outward from the heart of the model, the Arduino UNO microcontroller. The circuit's brain provides power to the small controller boards such as the L298N and KY-004 and dispatches signals to the motor drivers. The Arduino UNO is powered by the Power Supply Unit (PSU) and is connected to an Arduino CNC Shield to drive a stepper motor, as shown below.

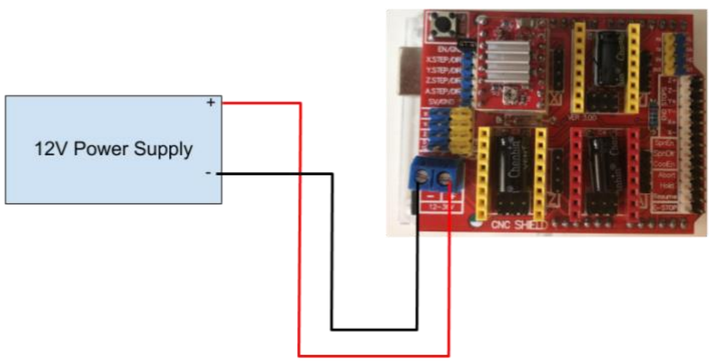


Figure 21: Power Supply Charging the Arduino

As the Arduino was covered up by the Arduino Shield, the team had to wire the components to the Arduino Shield, which itself was linked to the Arduino. The Arduino Shield was connected to the PSU's 12V cable, and this voltage would also be sent to the stepper motor. Since there was only one 5V pin on the Arduino Shield board, its 5V and GND pins had to be connected to a Mini Breadboard kit to power the L298N motor controller board, KY-004 push-button sensor, and the Ultrasonic sensor, as displayed in Figure 22.

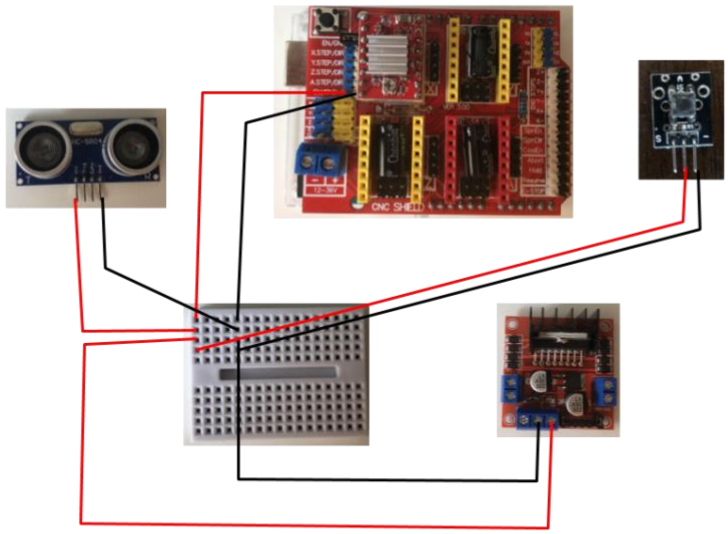


Figure 22: Power Different Parts Using a Mini Breadboard Kit

For the stepper motor, when NEMA 17 was used, the team only needed to plug the motor's cables to the x-axis pins' headers to make the motor function properly. The brushed DC motor was mounted to the left-half side of the motor controller as seen in Figure 23, and in turn connected to the Arduino shield, so are the two sensors.

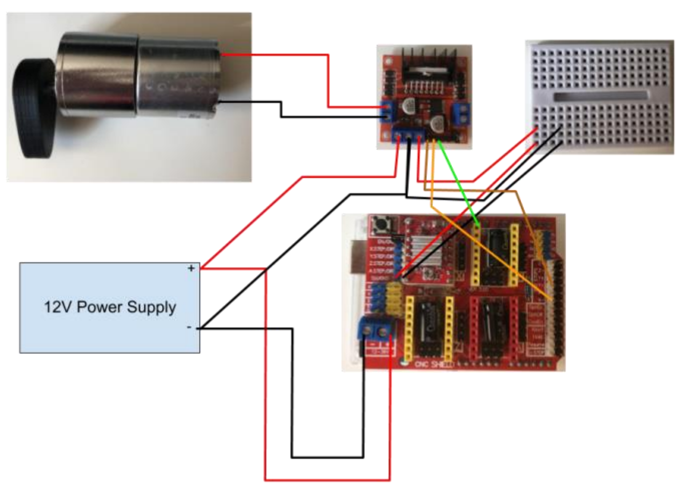


Figure 23: Connect Brushed DC Motor and Arduino Together with a Motor Controller

the team had to follow schematics of Arduino Shield CNC and find the corresponding signal pins on the diagram and then connected those pins with the trigger and echo pins on the ultrasonic sensor, signal pin on the KY-004 as well as ‘EnA’, ‘In1’, and ‘In2’ pins on the L298N, as shown in Figure 24.

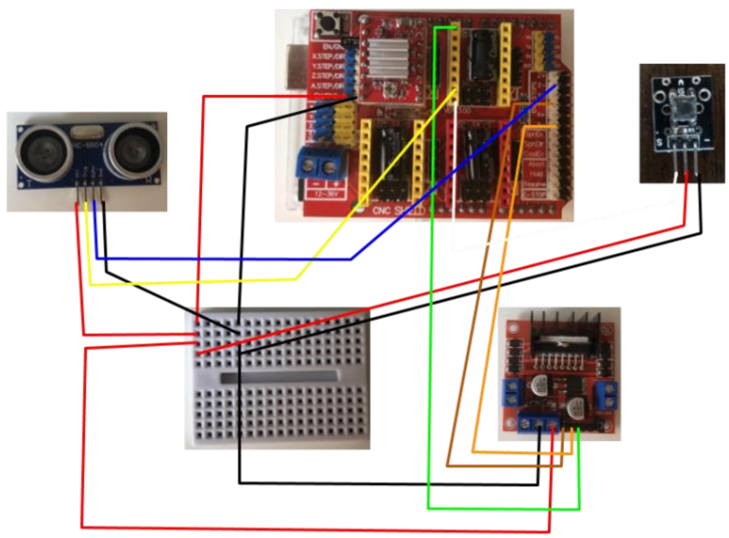


Figure 24: Connect All the Parts Together Using Arduino Shield

The team used half of the input pins on the L298N because the project only uses a single DC motor. The left-side pins were the input pins for possible other DC motors. The final step in completing the prototype was connecting the 12V and GND wires of the power supply to the 12V and GND pins on the L298N and the two output pins to the brushed DC motor to provide power for the motor to rotate. The final schematic of the complete circuit is shown below.

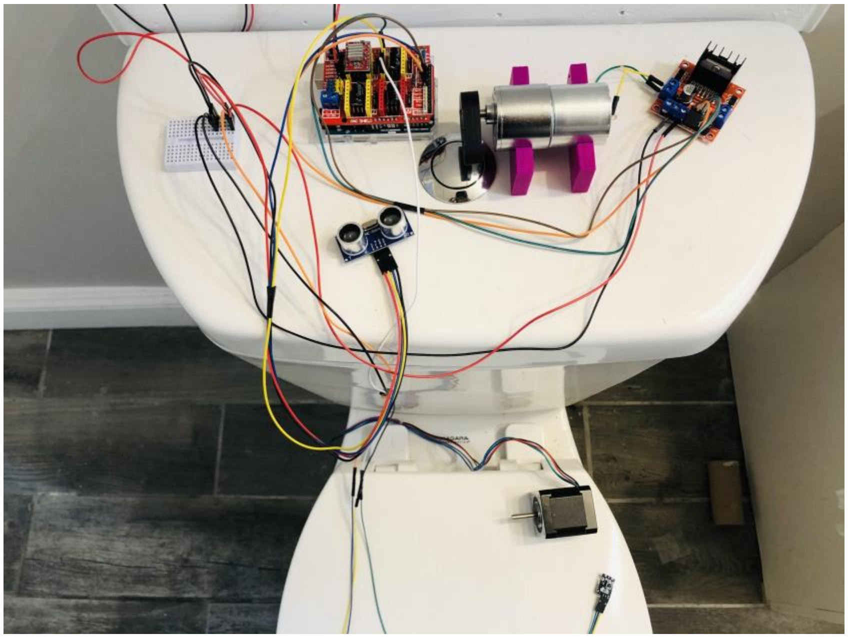


Figure 25: Final Layout of the Complete Circuit

6.4.1.2 What the Team Realized

When the team was developing the design solution, an outline of the installation and testing procedures was also created. After the circuit was assembled, the team intended to install the device to one of the toilets to test if the mechanism worked. This prototype would have had the Arduino with an Arduino Shield, the brushed DC motor, its motor controller board, and the ultrasonic sensor mounted to the top of the toilet's water tank. These components were placed under a wooden rectangular box. The wooden box was designed in SolidWorks and the panels were manufactured using the accessible laser cutter. The panels were then glued together. As can be seen in Figure 26, the team has assembled the series of components that were expected to install under the wooden cover, with the exclusion of the PSU. The 3D printed components, such as the teardrop shaped peg and the motor mount were manufactured using the TA's 3D printer. The team has successfully realized as much of the prototype as was possible.

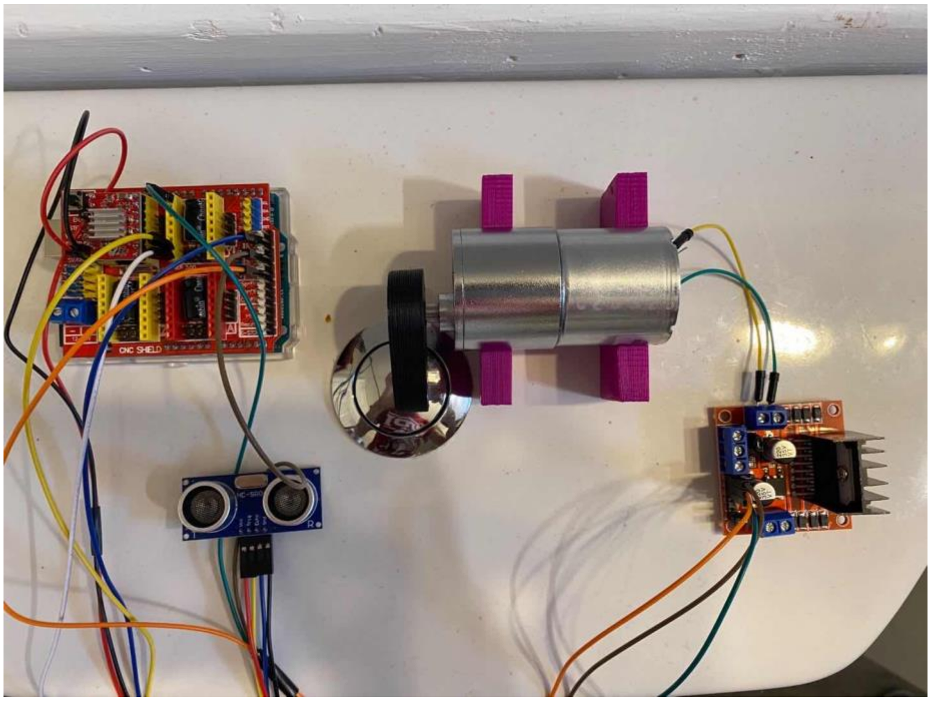


Figure 26: Toilet Installation Realization

6.4.1.3 What Was Tested/Modeled in SolidWorks

As the team members with the prototype construction supplies lived in rental rooms, they lacked permission from their landlord to make any physical modifications to the toilet. The installation of the pushing button sensor and the stepper motor instead, would be emulated and demonstrated in the SolidWorks software.

6.4.1.4 Device Testing

The PSU was not covered by the wooden box, so that the team could manually control the power, for safety reasons. The team tested their bare circuit, including the PSU and Arduino Shield, and developed a short video to demonstrate that the circuit fulfilled the team's expectations.

6.4.2 Code for the Complete Circuit

The software for the complete circuit could be divided into three separate sections. The first part, as displayed in Figure 27 contains the definitions of the pin number they used and setting up the pin mode to ‘INPUT’ or ‘OUTPUT’. In this section, they defined parameters that they would utilize in the following lines of code. For instance, the parameters ‘duration’ and ‘distance’ were selected to serve as numbers to calculate the distance between the operator’s hand and the ultrasonic sensor.

MQP_Final_Logic

```
// Final Logic Code for our MQP

// defines pins numbers
// DC Brush Motor
#define EnA 10 //L298N.EnA brown
#define In1 9 //L298N.In1 orange
#define In2 8 //L298N.In2 green

// Stepper Motor
const int stepPin = 2; // X.STEP
const int dirPin = 5; // X.DIR

// Ultrasonic Sensor
const int trigPin = 3; // Ultrasonic.trig  yellow
const int echoPin =  11; // Ultrasonic.echo blue


// Push Button Sensor
const int buttonPin = 6; //Push button pin white

// defines variables
// Push Button Sensor
int buttonState; // a numeric variable to the button state

// Ultrasonic Sensor
long duration; // duration of sound's going back and forth
int distance; // the detected distance

void setup() {
  // put your setup code here, to run once:
  pinMode(stepPin,OUTPUT);
  pinMode(dirPin,OUTPUT);
  pinMode(EnA, OUTPUT);
  pinMode(In1, OUTPUT);
  pinMode(In2, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication

}
```

Figure 27: Part#1 of the Final Logic Program

In the second piece of the control software, the team defined two functions Flush() and LidClose() outside the final loop section to make the logic more apparent and more accessible for modification. The first function 'Flush()' would command the brushed DC motor to rotate and push the flush-button on top of the water tank. After the flushing, the current flowing through the two wires of the brushed DC motor would be set to 'low' and turn off the motor. The function 'LidClose()' would order the stepper motor to lower the lid. This function does not pass any information to the other functions in the software.

```

void Flush() // drive the DC brush motor to flush the toilet
{
  // turn on the DC brush Motor
  digitalWrite(In1, HIGH);
  digitalWrite(In2, LOW);
  // set speed to 150 out 255
  analogWrite(EnA, 200);
  delay(4000);

  //turn off the motor
  digitalWrite(In1, LOW);
  digitalWrite(In2, LOW);
}

void LidClose() // drive stepper motor to close the lid
{
  digitalWrite(dirPin,HIGH); // Enables the motor to move in a particular direction
  // Makes 50 pulses for making one 1/4 cycle rotation
  for(int x = 0; x < 50; x++) {
  digitalWrite(stepPin,HIGH);
  delayMicroseconds(500);
  digitalWrite(stepPin,LOW);
  delayMicroseconds(500);
 }
 delay(1000); // One second delay
}
```

Figure 28: Part#2 of the Final Logic Program

The software's final section contains the program's most essential commands. The characteristic Arduino software "void loop ()" function serves as the code's executing body. This function acts as an infinite loop. This function orders the ultrasonic sensor to continuously record the distance between itself and objects located ahead unless the distance between them is larger than 10 centimeters and less than 50 centimeters. Forty centimeters is the range that the team estimates a person's hand will be from the ultrasonic sensor.

The loop function utilizes built-in logic functions such as ‘if ()’ functions to control the ultrasonic sensor. When the ‘if (buttonState == LOW)’ function is satisfied, the button sensor will inform the Arduino that the toilet lid is already down. Now that the Arduino knows that it is safe to flush the toilet it will engage the flushing software function. Similarly, then the ‘if (buttonState == HIGH)’ is satisfied, the seat lid closing function will be engaged, and the toilet will be flushed. After the waste has been flushed, the ultrasonic sensor will wait two seconds before it begins to record distances again. The system is now back into its standby mode.

MQP_Final_Logic

```

void loop() {
  // First Step: Collect the distance
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance
  distance= duration*0.034/2;
  Serial.print("Distance: ");
  Serial.println(distance);

  // Second Step: Identify if someone want to flush the toilet
  if (distance >= 10 && distance <= 50){
    buttonState = digitalRead(buttonPin);

    // Third Step: Check if the lid is closed and flush the toilet
      // first case: lid is closed
    if (buttonState == LOW){
      Flush();
      delay(2000);
    }
      // Second case: lid is not closed
    else if (buttonState == HIGH){
        // Close the lid
        LidClose();
        // Then flush the toilet
        Flush();
        delay(2000);
    }
  }


}
```

Figure 29: Part#3 of the Final Logic Program

7.0 Results and Conclusions

With each iterative improvement, the circuits and software were evaluated to determine whether the team could move onto the next stage of the design journey. Each successful test improved the confidence that the team could complete the design solution before the time limit expired. In the end, a piece of technology was successfully developed that benefited the communities.

As the world continues to spiral into disarray because of the Novel Coronavirus, there is an increasing public awareness about the need to prevent the transmission of diseases in shared public spaces. The project team embarked on a quest to realize a universally applicable smart toilet appliance that would offer the most impact upon shrinking the spread of harmful bacteria and viruses. Using commercially available small-scale electrical components, such as those found in an Arduino Starter Kit, they demonstrated the ability to create entrepreneurial solutions to the world's problems.

Having set out to design and assemble a device that could integrate into any type of toilet and eliminate bacteria, the team narrowed the solution to an appliance that shut the lid on a domestic toilet before flushing. The benefit of the current design is that it can be installed on numerous domestic toilets and prevent the spread of the current pandemic in homes. Without access to Worcester Polytechnic Institute (WPI)'s woodshop with its laser cutter or any other machine workshop, the team could not manufacture all of the cover components that the team had earlier envisioned. The realized deliverable was demonstrated, as seen in Figure 30, and recorded as a video, but the temporarily inaccessible half of the demonstration was simulated in SolidWorks. This SolidWorks simulation was developed as a video. Both demonstrations can be accessed using the link given in Appendix B.

Figure 30: The Final Water Flush Device

Despite the changes the project team has been forced to make, the project remains a successful solution to limiting the spread of harmful bacteria and viruses from the toilet bowl.

8.0 Future Development

After pursuing this project for eleven weeks, the team has completed the project solution for a household smart toilet appliance that helps to prevent the spread of harmful bacteria. The limitations of the timetable and the setbacks that the team endured led the team to scale back the design goals. This smart toilet appliance can be improved. For future Major Qualifying Project (MQP) teams to continue the work, the team recommends the following to be considered.

8.1 Ultraviolet Lights that Kill Bacteria

Ultraviolet (UV) lights can be implemented into the design to destroy the bacteria and viruses that reside in the toilet bowl. UV lights utilize very short wavelengths to generate large amounts of heat to kill bacteria. These UV lights will result in a higher voltage drain on the PSU, and the current PSU might not be able to output enough energy to power the entire appliance. The UV lights will also need to be waterproof to survive inside the toilet bowl. New mounts will also need to be created to install the lights. Naturally, new and longer cables will need to be procured [27].

8.2 Mass Manufacturing and Installation

A truly entrepreneurial solution to the spread of COVID-19 and other viruses from toilet bowls would be mass manufacturable and versatile. The initial concept was meant to meet these lofty goals; however, as deadlines approached and such considerations because of secondary importance to a basic prototype’s completion, the project team scaled back the idealism. The team has not abandoned the vision. The components that the team selected to manufacture the prototype may not be suitable for a mass-produced product. While the prototype consisted entirely of off-the-shelf and affordable parts, the availability of such components may require an extensive redesign of the project solution. Similarly, adding the ability to operate a commercial toilet handle would allow for greater flexibility in the toilet appliances applicability. This feature would also require extensive modifications to the hardware and software. The design needs extensive calibration on every commercial toilet because there is no uniform standard of force set to operate these handles. Since the feature's concept requires the use of a durable metal cable, care must be applied to ensure the customized length of wire does not lead to structural failure and potential harm to any toilet operator.

8.3 Installation Procedures

As the project's time limit neared its expiration, the project team looked for materials and solutions that could be quickly applied to solve the troubles of installing the various components onto a residential toilet. Tape, duct tape and electrical tape, increased the ease of prototyping since it was cheap, immediately available in the homes, waterproof, and removable. The team also managed to build some 3D parts using the Teaching Assistant (TA)'s 3D printer, such as the toilet flushing teardrop peg. The team lacked access to any sort of workshop or precision power tools. Working with the toilet seat mounting brackets' limited surfaces, the team chose not to use an electric power drill because it would remove too much material from the plastic brackets. Large drill heads would compromise the structural integrity of the plastic brackets. The team also lacked access to a manufacturing facility to build a thin metal rod long enough to thread between the holes in the brackets. A future project team would have to determine a metal with the optimum material characteristics for the metal rod used in the design as well as a suitable method to drill into the fragile toilet seat brackets.

8.4 How the Design Could be Simplified for Mass Production

The team has utilized plenty of 3D printing and handcrafted components and covers in the prototype's construction. These covers are affordable and straightforward to construct the prototype but are not ideal for mass manufacturing. Large numbers of individually crafted 3D printed parts are not ideal for economies of scale. Depending on the printers contracted, such parts might not be economical since any additional manufacturing costs are passed onto the consumer.

Modifying the smart toilet appliance for mass manufacture, the team recommends that the cover's construction material. Altering the construction material from wood and PLC to a type of sheet metal would ease manufacturing and improve durability since these parts can be manufactured with machine tools. Plastics offer weight savings, which place less strain on a toilet; however, designing plastic molds is difficult and expensive. The assembly cost may increase because of the need for experienced machinists and metals. The circuit board can be replaced by a smaller PCB that is easier to fit into a casing.

8.5 Alternate PSU

The Power Supply Unit (PSU) that the project team pursued was chosen based on the Teaching Assistant (TA)'s experience configuring it. One of the technical advisors, Leo Gross, an electrical engineering student at Worcester Polytechnic Institute, disagreed with the choice for a PSU. He insisted that the ATX power supply will not work well under minimum electrical load conditions. If the circuit design did not place enough current draw on the PSU, then the circuit and the motors would remain unpowered. This limitation meant that another external power supply would be needed. The circuit design avoided this complexity by carefully wiring the components. The PSU required a signal to control the power supply. The team did not wire the PSU in such a manner that it would require an electrical command from the Arduino. It is wired such that it operates like a conventional battery. The current PSU is bulky and heavy. A lighter and cheaper PSU that requires less complex wiring would be better suited for a mass-manufactured product. Only a mass-manufactured product would have an impact on the current pandemic and future communal health.

8.6 A Solenoid to Push Handle on Commercial Toilet

An additional mechanism to push a side-mounted flush handle is necessary to add flexibility to the smart toilet appliance's design. A solenoid would have sufficient energy to force the toilet flush down yet still retract the flush handle back. Solenoids are powerful mechanisms; however, they can be heavy, complex, slow, and expensive, characteristics unideal for a mass-manufactured appliance. Despite these drawbacks, a solution to increase the compatibility with domestic and commercial toilets must be added to the existing design.

8.7 Ensuring a Long Service Life

The project team has conducted extensive exercises to determine the simplest yet effective method of insulating the electronic components from the moisture that surrounds the toilet bowl. The solutions during the waning hours before the project due date were casings, coverings, and electrical tape. A mass-manufactured product needs its parts to withstand moisture for much more extended periods; these current solutions are not acceptable. Moisture protection will remain a tough problem to solve. The ultrasonic sensor that notices activity to start the safe flushing sequence cannot be adequately covered up. Transparent plastic wrappings or other thin covers might keep the moisture out but will result in false returns. A future project team will need to focus considerable energy to solve the problem of moisture damage.

Appendices

Appendix A

The mark of an Engineer is the success in solving difficult problems efficiently. The team of Mechanical Engineers has worked diligently to solve every problem that has come, including problems outside of the field of study. Not every problem that was encountered was immediately solved. Outside help was sometimes called in to keep the project's momentum.

Appendix A.1- Bad Power Supply

Right at the beginning of the project, the team encountered the first problem: how to power the Arduino Uno. The first solution was to use a USB cable connected to a computer. The USB cable offered a temporary solution. The team needed to find a superior alternative. They extensively researched different kinds of batteries, often focusing on lithium-ion battery types. With help and guidance, the team identified a Power Supply Unit (PSU). However, the initially ordered and tested PSU was not providing the amount of voltage that it was designed to do. The team concluded that the PSU was faulty and ordered a replacement. This set back the rate of progress by at least a week.

Appendix A.2- Incorrect Wiring

As the group tried to test the stepper motor, a stepper motor driver and a capacitor were wired to the circuit board. However, due to oversight and reversely wired positive and negative sides of a capacitor, a small explosion occurred and the capacitor was destroyed. This experience educated the team that they must make sure that the circuit was correctly set up before connecting it to the power connection.

Appendix A.3- The Ground Connection Issue

In a manner similar to the earlier wiring mistake the team did not connect the PSU's ground wire to the Arduino ground pin. The team members noticed the smell of burning plastic and powered down the PSU. A series of copper wires had been burnt as a result of this mistake.

Appendix A.4- Unfamiliar with the Arduino's Software

As was mentioned in Chapter 3, the team consisted of four mechanical engineering students with limited experience and background knowledge in electrical and computer science. Moreover, the only student who had worked with the Arduino Uno before was not able to work on the complete circuit due to the fact that he was not able to meet the remaining team members in Worcester. The team members needed to learn how an Arduino functioned and how to develop software for it before they could effectively make design decisions. This learning period caused further delays. Eventually, the team developed sufficient expertise on Arduinos to complete the project.

Appendix A.5- Problems with Component Delivery

Throughout this project, the team has experienced delayed or incorrect component deliveries several times. The first instance of this progress retarding issue was when the PSU arrived without the necessary wires to connect it to any of the other components. The second when the team ordered several new parts that were then stolen. The parts did not arrive when the delivery service notified that the parts would be at the door. Consequently, the team was not able to accept the delivery in person and the parts became unrecoverable. Finally, when the team tried to order the longer wires, an unknown type of cable arrived. Eventually, the necessary parts would arrive and the project could be finally finished before its due date.

Appendix B

To present a more intelligible scene of how the designed device worked, the project team recorded two videos, each separately showing different kinds of realizations. Both videos were uploaded to YouTube and links are provided in the following sections.

Appendix B.1- Video of the Water Flushing Part

In this video, the mechanism inside the plywood box was presented, and the team activated the system to demonstrate how the brushed DC motor with the bearing flushed the toilet. The video highlights a key project constraint. The team members resided in a rental apartment and could not make permanent alterations to the toilet seen in the video. Rather than fixing all the consonants to the toilet with glue, the team held the motor to realize the demonstration. If the smart toilet appliance was installed onto the toilet, the parts would be fixed in similar positions as demonstrated in this report. The link is here:

https://www.youtube.com/watch?v=JuPxBl6X8rg

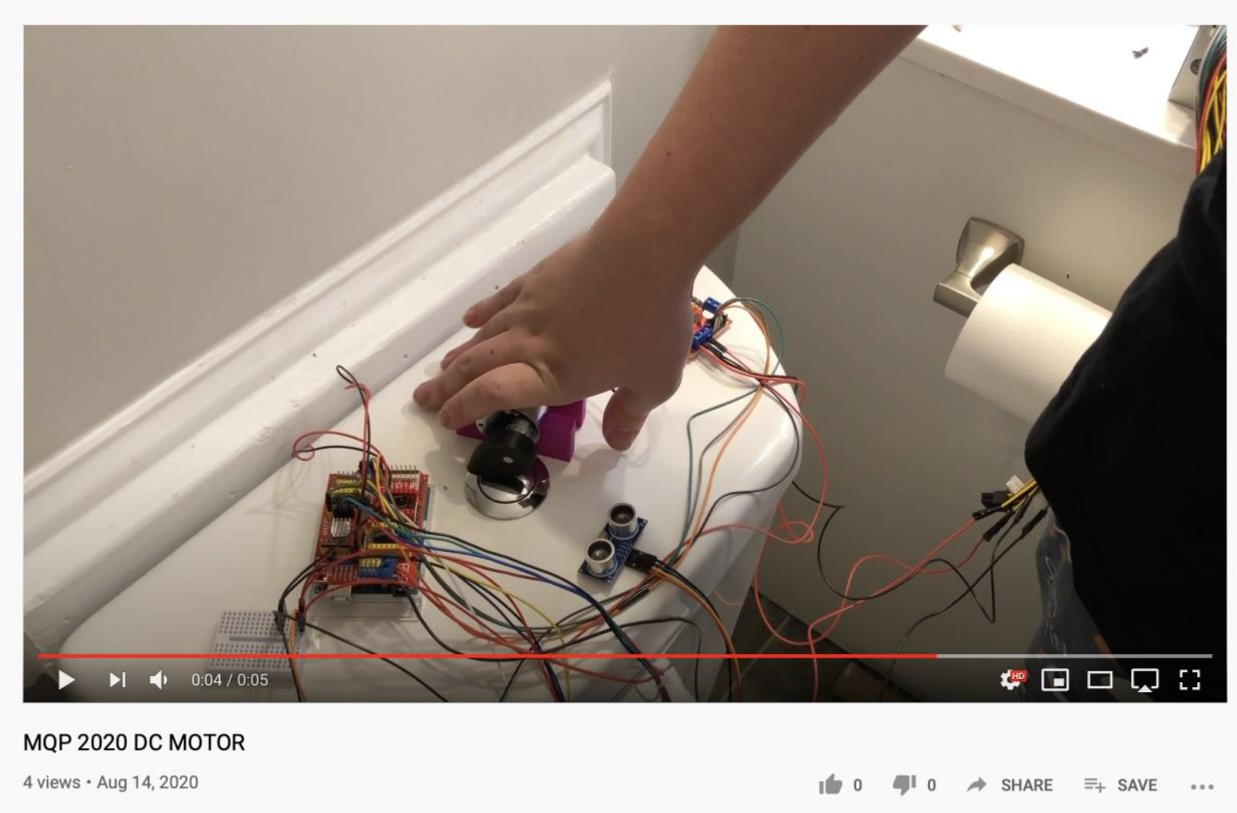


Figure A1: Video of the Water Flushing Part

Appendix B.2- Video of the Lid Closure Part

In this video, the working process of the lid-closure part was demonstrated. Unlike the previous video, here all of the components are placed in their intended location upon a simulated toilet. A shaft is drilled through the toilet seat and toilet lid which connects to the stepper motor. A pad to depress the button sensor is also fixed to the toilet lid. As SolidWorks cannot simulate external computer programs such as the Arduino software, the team created a motion capture video that illustrates the motion a real toilet fitted with the smart toilet appliance will look and behave. The link to the video is here:

https://www.youtube.com/watch?v=IlTIel4inH4

Figure A2: Video of the Lid Closure Part

Appendix C- Code to Realize the Water Flushing and Lid Closure Process

The software program utilized to control the Arduino microcontroller was the heart of this project. This code's design was discussed in Chapter 6.4.2. Please view this chapter for details. The software used to command the smart toilet appliance lies here:

```
// Final Logic Code for the MQP

// defines pins numbers
// DC Brush Motor
#define EnA 10 //L298N.EnA brown
#define In1 9 //L298N.In1 orange
#define In2 8 //L298N.In2 green

// Stepper Motor
const int stepPin = 2; // X.STEP
const int dirPin = 5; // X.DIR

// Ultrasonic Sensor
const int trigPin = 3; // Ultrasonic.trig  yellow
const int echoPin =  11; // Ultrasonic.echo blue

// Push Button Sensor
const int buttonPin = 6; //Push button pin white

// defines variables
// Push Button Sensor
int buttonState; // a numeric variable to the button state
```

```
// Ultrasonic Sensor
long duration; // duration of sound's going back and forth
int distance; // the detected distance

void setup() {
  // put your setup code here, to run once:
  pinMode(stepPin,OUTPUT);
  pinMode(dirPin,OUTPUT);
  pinMode(EnA, OUTPUT);
  pinMode(In1, OUTPUT);
  pinMode(In2, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
}

void Flush() // drive the DC brush motor to flush the toilet
{
  // turn on the DC brush Motor
  digitalWrite(In1, HIGH);
  digitalWrite(In2, LOW);
  // set speed to 150 out 255
  analogWrite(EnA, 200);
  delay(4000);

  //turn off the motor
  digitalWrite(In1, LOW);
  digitalWrite(In2, LOW);
}
```

```
void LidClose() // drive stepper motor to close the lid
{
  digitalWrite(dirPin,HIGH); // Enables the motor to move in a particular direction
  // Makes 50 pulses for making one 1/4 cycle rotation
  for(int x = 0; x < 50; x++) {
  digitalWrite(stepPin,HIGH);
  delayMicroseconds(500);
  digitalWrite(stepPin,LOW);
  delayMicroseconds(500);
 }
 delay(1000); // One second delay
}

// pushing button sensor is activated

void loop() {
  // First Step: Collect the distance
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance
  distance= duration*0.034/2;
```

```
  Serial.print("Distance: ");
  Serial.println(distance);

  // Second Step: Identify if someone want to flush the toilet
  if (distance >= 10 && distance <= 50){
    buttonState = digitalRead(buttonPin);

    // Third Step: Check if the lid is closed and flush the toilet
      // first case: lid is closed
    if (buttonState == LOW){
      Flush();
      delay(2000);
    }
      // Second case: lid is not closed
    else if (buttonState == HIGH){
        // Close the lid
        LidClose();
        // Then flush the toilet
        Flush();
        delay(2000);
    }
  }
}
```

Bibliography

[1] W. Wang *et al*, "Virology," *Virology,* vol. 474, pp. 19-27, 1955. Available: http://www.sciencedirect.com/science/article/pii/S0042682214004723.

[2] Anonymous (). *WHO | Summary of probable SARS cases with onset of illness from 1 November 2002 to 31 July 2003*. Available: https://www.who.int/csr/sars/country/table2004_04_21/en/.

[3] D. S. Hui *et al*, "The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health — The latest 2019 novel coronavirus outbreak in Wuhan, China," *International Journal of Infectious Diseases,* vol. 91, pp. 264-266, 2020. Available: https://www.ijidonline.com/article/S1201-9712(20)30011-4/abstract. DOI: 10.1016/j.ijid.2020.01.009.

[4] WHO. (17 April). *Q&A on coronaviruses (COVID-19)*. Available: https://www.who.int/news-room/q-a-detail/q-a-coronaviruses.

[5] C. P. Gerba, C. Wallis and J. L. Melnick, "Microbiological hazards of household toilets: droplet production and the fate of residual organisms," *Journal of Applied Microbiology,* vol. 30, *(2),* pp. 229-237, 1975. Available: https://arizona.pure.elsevier.com/en/publications/microbiological-hazards-of-household-toilets-droplet-production-a.

[6] (). *Coronavirus in the U.S.: Latest Map and Case Count*. Available: https://www.nytimes.com/interactive/2020/us/coronavirus-us-cases.html.

[7] J. Barker and M. V. Jones, "The potential spread of infection caused by aerosol

contamination of surfaces after flushing a domestic toilet," *Journal of Applied Microbiology,* vol. 99, *(2),* pp. 339-347, 2005. Available: https://sfamjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2672.2005.02610.x. DOI: 10.1111/j.1365-2672.2005.02610.x.

[8] Anonymous (-03-23T12:39:29-04:00). *Study suggests new coronavirus may remain on surfaces for days*. Available: https://www.nih.gov/news-events/nih-research-matters/study-suggests-new-coronavirus-may-remain-surfaces-days.

[9] (). *5 Germs You Really Can Get From a Toilet*. Available: https://health.howstuffworks.com/diseases-conditions/infectious/5-diseases-toilet3.htm.

[10] M. A. Said, T. M. Perl and C. L. Sears, "Healthcare epidemiology: gastrointestinal flu: norovirus in health care and long-term care facilities," *Clin. Infect. Dis.,* vol. 47, *(9),* pp. 1202-1208, 2008. Available: https://pubmed.ncbi.nlm.nih.gov/18808354/. DOI: 10.1086/592299.

[11] Geneva M. Wilson. (). *Flushing toilets may spread bacteria in hospitals*. Available: https://www.healio.com/news/infectious-disease/20200213/flushing-toilets-may-spread-bacteria-in-hospitals.

[12] Lexamer. (). *Hand with magnifying glass showing bacteria in the toilet bowl, wc hygiene vector concept*. Available: /image-vector/hand-magnifying-glass-showing-bacteria-toilet-599998142.

[13] Anonymous "A Brief History Of The Toilet And Other Potty Trivia," 2016. Available: https://thetoiletseatguy.com/a-brief-history-of-the-toilet-and-potty-trivia.

[14] SHAUNACY FERRO. (-06-16T09:30:00+00:00). *Why Are Public Toilet Seats U-Shaped?*. Available: https://www.mentalfloss.com/article/64677/why-are-public-toilet-seats-u-shaped.

[15] (09-Feb-). *How Tankless Toilets Work* . Available:

https://home.howstuffworks.com/tankless-toilet.htm#pt3.

[16] JACK GERARD. (). *How Sloan Flush Valves Work*. Available: https://www.hunker.com/13400991/how-sloan-flush-valves-work.

[17] Automatic toilet flushing system, John T. Barrett. Available: https://patents.google.com/patent/US4756031A/en).

[18] Auto flush for tank toilet, Bennie N. Veal. Available: https://patents.google.com/patent/US5603127A/en).

[19] Toilet assembly having an automatic ventilation system, Jae K. Sim. Available: https://patents.google.com/patent/US5715543A/en.

[20] Anonymous (31-Jan-). *NEVER FORGET TO CLOSE THE TOILET SEAT AGAIN!*. Available: https://flushdowntoiletseat.com/.

[21] T. Lee, R. Lee and P. Liu, "Design and production of toilet seat lifting mechanism," *MATEC Web Conf.*, vol. 71, pp. 04010, 2016. Available: https://www.matec-conferences.org/articles/matecconf/abs/2016/34/matecconf_ccpe2016_04010/matecconf_ccpe2016_04010.html. DOI: 10.1051/matecconf/20167104010.

[22] (). *Brush DC Motor Guide*. Available: https://www.anaheimautomation.com/manuals/forms/brush-dc-motor-guide.php.

[23] Bill Earl. (May 05,). *What is a Stepper Motor?*. Available: https://learn.adafruit.com/all-about-stepper-motors/what-is-a-stepper-motor.

[24] Popular models | 3D CAD Model Collection | GrabCAD Community Library. Available: https://grabcad.com/library.

[25] Multisim. (). *Multisim Live Online Circuit Simulator*. Available: https://www.multisim.com/.

[26] B. Kruger, "Arduino-CNC-Shield-Schematics," 2013. Available: https://blog.protoneer.co.nz/arduino-cnc-shield/arduino-cnc-shield-schematics/.

[27] J. Cooper *et al*, "Efficacy of an automated ultraviolet C device in a shared hospital bathroom," *Am J Infect Control,* vol. 44, *(12),* pp. 1692-1694, 2016. Available: https://pubmed.ncbi.nlm.nih.gov/27575773/. DOI: 10.1016/j.ajic.2016.07.004.