SHTTL

Author: Manny Hur, Xander Li, Joanna Wickersham: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—A device capable of accurately measuring and dispensing spices to simplify use and cleanup during the cooking or baking process.**

***Index Terms*—Design, Dispense, Fan, IR Sensor, Motor, Raspberry Pi, PCB, Spices**

I. Introduction

For many, cooking and baking can be a daunting task for reasons beyond skill. It can be a messy process and one that is hard to do accurately, especially in the case of baking. Thus the problem we are trying to solve is that there is no portable, organized way to get accurate measurements. The goal of SHTTL is thus to simplify the process of cooking in each of these domains. It focuses mainly on hardware applications, with some software components. This product is useful for anyone who cooks frequently at home and/or is looking to consolidate their most commonly used spices. There are many solutions available for either storing spices in an organized fashion, but there are no good portable solutions that will also accurately dispense spices for you. With our device, one can simply hold the device over, say, a mixing bowl and dispense with a touch of a few buttons. Our main goals, based on the goals of users, are to focus on accuracy and portability. In order to test the success of the solution, we will judge size, accuracy, and speed. We intend to have volumetric dispense accuracy of +/- 10%. The spray radius of the device will be less than 6" in diameter when 6" from target. The dimensions of the device will be less than 4"x4"x8" and will weigh less than 500g to provide portability. Other metrics of focus will be charging speed (less than five hours) and battery life (2 days with nominal use).

II. Design Requirements

For our MVP, we expect our device to be fully functional for 1-2 spices. The dimensions of the device are 4"x4"x8" and weigh under 500g when empty. We decided on the weight metric because it is equivalent to two iphones, which will ensure its portability and ease-of-use. A similar case is made for the size, which we based off of a Starbucks venti cup, another obviously portable dimension.

Users will be able to dispense their desired spice with the push of a button and use a potentiometer to choose the amount of spice to be dispensed. Spice dispense accuracy must be within 10% of target measurement. This is a requirement because the intended use case for the device would be for cooking and following specific measurements in recipes; the accuracy of the measurement of the dispensed spice must be reasonable for the device to be practical in a cooking setting. We will test this metric by requesting a specific measurement to be dispensed and then measure the resulting amount dispensed with with measuring spoons. The volume of spice remaining, which is measured by the IR distance sensor, must also be within 10% of the actual volume remaining so the user at least has a good idea of when it is time to restock on a spice. We can test this metric by comparing the measured value to the actual volume of the spice remaining, and tweak the raspberry pi code to improve accuracy. It is difficult to be very accurate because the IR distance sensor only measures the distance between the top of the compartment to the surface of the spice, which may not be uniform.

The device is also designed to help evenly distribute the spice amongst the target with a spray radius. The spray distance from the dispense mechanism will have a 6" diameter from the target location when the device is 6" above the target with a maximum diameter of 12" for larger heights and a minimum diameter of 2" for smaller heights. This will allow for an even dispersion of the spice amongst whatever the user is seasoning and ensure the user that no spice will spill if it is used properly given this information. We will test the dispense mechanism with a series of trials measured over a white piece of paper placed on a scale, and we can adjust the fan speed to achieve the desired metric.

The brevity of the use case also must be considered. The dispense speed will be timed and is expected to be under 1 tsp/sec in order to assure the user will not have an unreasonable idle waiting period (as this would defeat the purpose of simplifying the cooking process). We can test this by measuring the time it takes to dispense a fixed amount of volume, and we can adjust the motor speed to achieve the target metric. We will require a jamming rate of no more than once per month, monitored over the month of testing.

Since this device is wireless and rechargeable, the time it takes to fully charge and drain must be considered for the kitchen appliance to be practical. The device will be able to achieve a full charge in less than five hours and will conservatively last 2 days with nominal use. These metrics were also chosen to ensure the product is useful, not inconvenient. We will test this through full drain and charge trials and tests with varied quantity testing (e.g. 10-100 uses per day). Finally we will aim for a dispense noise level of under 60 db. This metric is approximately equivalent to the noise level of a conversation in an office or restaurant space [3]. Each test that requires trials will be done in sets of 20-30 per testing session.

Qualitatively, we want the device to accurately dispense the desired spices with reasonable speed and accuracy. The other metrics are to confirm portability and be a convenient, low maintenance solution.

III. Architecture and/or Principle of Operation

Our overall architecture is oriented around the user standpoint. We tried to visualize the final product as much as possible. The product maximum dimension of our product is limited by the dimensions of the largest Starbucks Cup. The device is organized to three major sections, the Upper, Middle, and Lower assembly. We will introduce each in more details as well as the interfaces between them. Figure A shows an overall rendering of the device.

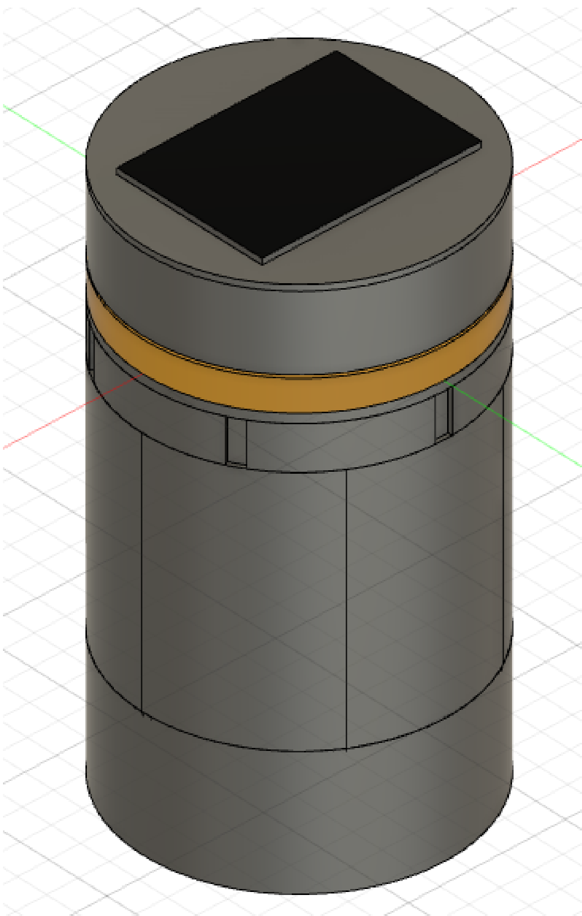


Fig. A Overall Rendering

This modular design provides better flexibility in designing each stage as well as redesigning them if needed. Modular design is also more convenient for our teamwork since we might not be all in Pittsburghs. Unit testing is also possible through this design, and we can attach different dispense mechanisms (see next section) to the bottom and switch them rapidly.

There are 2 ways users can provide input/actuate the device. The selection ring (in yellow), and buttons (will be attached to the vertical slots below the yellow ring). The selection ring can be used to select the desired dispense volume. There are 6 buttons, each corresponds to a spice. After selecting the desired volume, pressing the button will dispense the corresponding spice.

A. Hardware: Upper Assembly

The Upper Assembly, shown in Figure B, houses the control electronics, battery, and the selection ring. It is located at the top of the device in Figure A, and it includes every above the selection ring and the ring itself.

The selection ring (in yellow) is one of the ways users interact with Shttl. It rotates continuously in both directions. Users use it to navigate the menu or choose the volume that they want to dispense. The selection ring is held in place by four gears freely rotating on the base. The bigger gear has a continuous potentiometer attached to it (bottom left), with the back of the pot attached to the cross bar.

The electronics are also shown in the figure. The pink is the Pi, the red components are the motor controllers, the blue layer is the battery controller, and the grey rectangular cube is the battery. These electronics will be directly attached to the Pi as HATs. We left a generous amount of free space just in case the electronics do not stack as tightly as shown. We designed the base

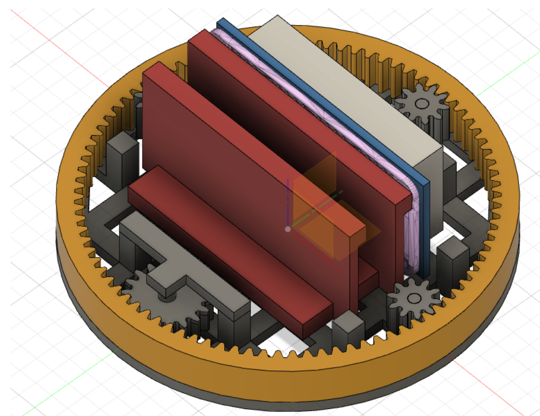


Fig. B Upper Assembly

B. Hardware: Middle Assembly

The Middle assembly houses our PCB, sensors, buttons, and fan. It is the layer with small vertical slots, located below the yellow ring in Figure B. Physically, it will be permanently attached to the Upper assembly. Electronically, we will have 2 by 20 ribbon cable connect the Pi and the PCB.

The fan will be attached to the center where there are four screw holes shown in Figure C. The 6 IR distance sensors (for measuring the remaining level of spices), will be attached to the cross bars supporting the fan. The fan is used to disperse the powder to a wider range. The PCB will be attached to the top of the fan, with a gap below to allow airflow. The flat tactile buttons will be attached to the vertical slots, and we will drill small holes on the wall for wires.

Not shown on the 3D model, we will have pogo pins along the circumference of the circle. Six groups of 2 pogo pins under six cross bars. These will be the connection to the DC motors on the lower assembly.

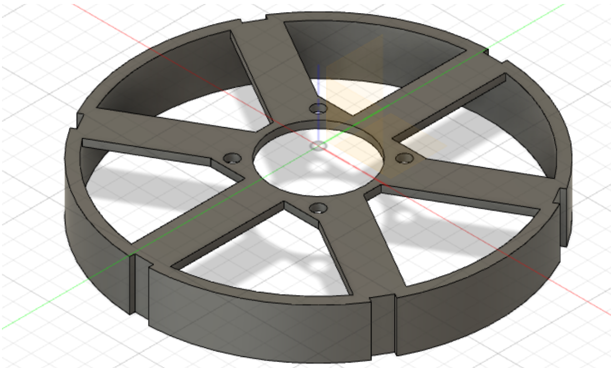


Fig. C Middle Assembly

C. Hardware: Storage Shaft and Lower Assembly

The lower assembly includes the storage shaft and the dispense mechanisms. It will connect to the Middle assembly via 3 pairs of latches (we have yet to design or decide the latch). The center holes is the airway, connecting the fan to the bottom where the airflow will blow the powder to a wider range than just dropping them.

The six square wire shaft will line up to the six groups of pogo pins, and the wires go to the bottom for DC motors. The Figure D has the outer shell of the dispense mechanism hidden to reveal the actual mechanisms. The details will be discussed in the next section. All possible dispense mechanisms will share the same opening, thus can be attached to the same storage shaft. For the final product, they will be permanently attached to the storage shaft.

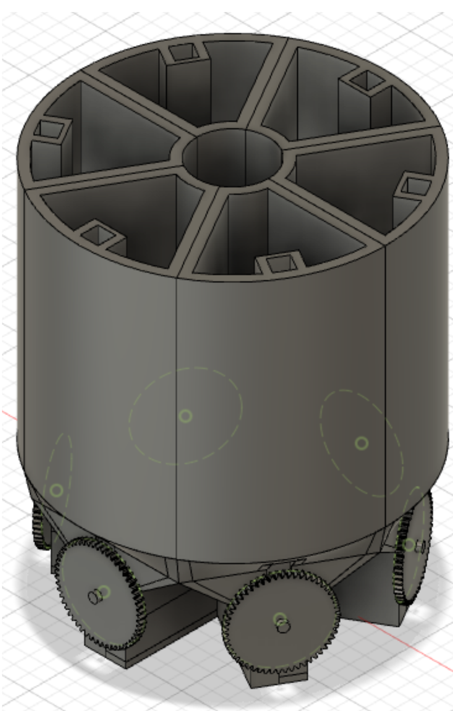


Fig. D Lower Assembly

Not shown in the model is a rocket engine bell shape nozzle at the bottom. The range of powder dispersion is affected by the nozzle shape, thus we can achieve the corresponding requirement by experimenting with different nozzle shapes.

D. System Architecture

Our system block (Figure 1) shows the relations between the different hardware components (Upper, Middle, and Lower assembly) and the software system. For the software, we have actuators, sensors, and software subsystems. The actuators subsystem includes motor controller bonnets located in Upper assembly, fans located in Middle assembly, and six DC motors in the lower assembly. The sensor subsystem includes the IR sensors and the tactile buttons located in the Middle assembly, and the Ring Selector located around the electronics. The software interaction between the subsystems will be discussed in detail in Section V. We organized the system into these three subsystems by their functionality.

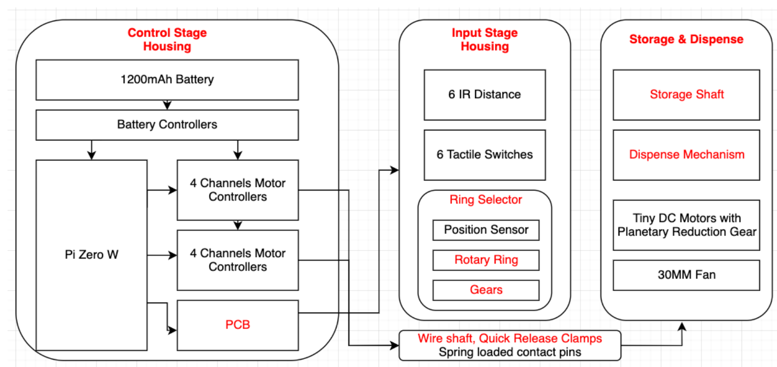


Fig. 1. Block Diagram, larger version at end of document on page 10

IV. Design Trade Studies

A. Dispense Mechanism

After researching existing designs, we realized that the dispense mechanism will be one of the most challenging aspects of Shttl. We drew inspiration from many power dispense mechanisms used in the packaging industry. During the adaptation to our scale, the accuracy is the biggest concern. We designed and considered three different dispense mechanisms. We need them “Rotary Cup Filler”, “Vertical Auger", and “Gumball Wheel”, as shown below in Figure E, F, and G.

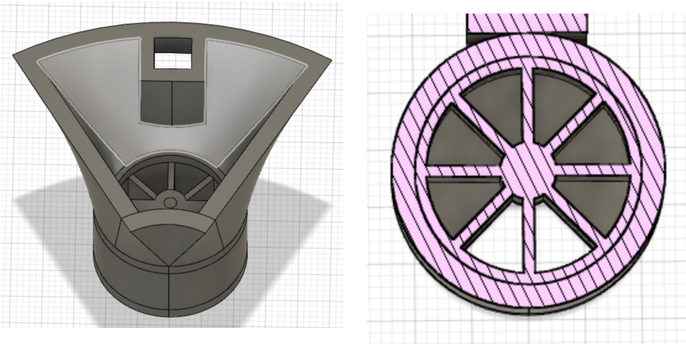


Fig. E Left: Rotary Cup Filler. Right: Cross section view from the top

The Rotary Cup Filler works by rotating the center rotor that has 8 sections. By offsetting the inlet port on the top and the outlet port on the bottom, we can achieve a very tight seal. The smallest volume we can dispense is determined by the volume of 1 section. The DC motor would be attached vertically to the back (below square hole in Figure E). The DC motor will have one gear attached to it, and it will drive another gear attached to the rotor at the bottom. This design allows flexibility in the trade-off between speed and accuracy, since we can easily adjust the size of the section. However, this circular sectioned rotor is also the biggest downfall of this design.

During testing, this design has the tightest seal when not used. It is nearly impossible to have powder dripping out when the motor is not rotating regardless how violent you shake it. However, the torque required by this design is simply too large for our DC motor. During testing, the rotor is extremely hard to move. We suspect that this issue is caused by the high contact area of moving parts. Bearing is considered and deemed useless since the majority of the contact surface is the wall structure. Lubrication is considered and deemed unfit since this area has direct contact with food. A slight variation of this design is also considered. We tried removing the circular wall. The fins (the straight line wall radially outward from center) are too long and not strong enough to drive the power. Shortening the fins will dramatically decrease the dispense speed to an unusable level.

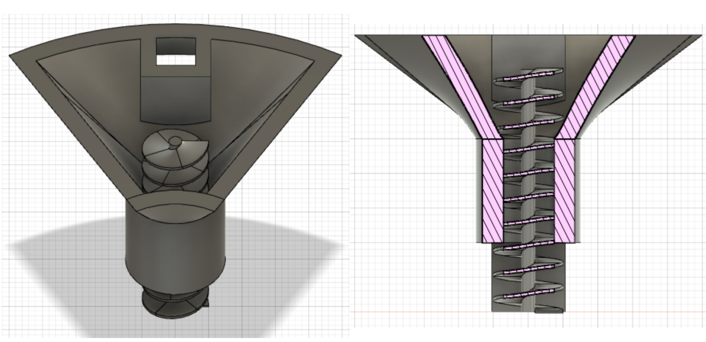


Fig. F Left: Vertical Auge. Right: Cross section view from the front

The second design we considered is the Vertical Auger, as shown in Figure F. This design is rather intuitive. When the auger rotates, the powder will be driven down by the auger. The DC motor would be attached vertically to the back (below square hole in Figure F). The DC motor will have one gear attached to it, and it will drive another gear attached to the rotor at the bottom. The Vertical Auger provides us the greatest control over accuracy. The angle of the auger spiral plates determine what the trade-off between what smallest control of volume can be and what level of dispense speed it can achieve.

During testing, we revealed another trade-off that the angle of plate affects. The flatter the spiral is, the more frictional force caused by the powder. The frictional force breaks off the axle as well as the spiral plate often. We experimented with thicker axles as well as a thicker spiral plate. However, thicker plates implies that the gap in between the plates are small, thus dramatically reducing the dispense speed to an unusable level. Another issue with this design is the production difficulty. We’re using a UV resin 3D printer for the production of these parts. For any 3D printers, in order to print a spiral, one has to add supports between the plates or split it in half down the axle and print it horizontally. The small gaps between the plate prevented us from using the first option. The auger total diameter is less than 10 mm. During the split and print method, we had a really hard time having accurate enough prints. Manual sanding is only used in a very limited way since the spiral plates are thin and the two halves have to line up perfectly.

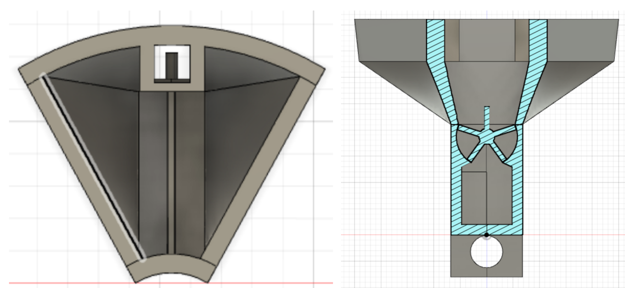


Fig. G Left: Gumball Wheel. Right: Cross section view from the front.

The third and the chosen option for Shttl is the Gumball Wheel design. This design is inspired by the cereal dispensing machine and gumball machine. As shown in Figure G, the Gumball wheel rotates and dispenses powder. The DC motor will be inserted to the circular hole at the bottom. The DC motor will have one gear attached to it, and it will drive another gear attached to the rotor at the bottom. At each moment, one of the six sections can dispense through the outlet. The smallest volume we can dispense is determined by the volume of the section. As a direct result, the length of the gumball wheel can be adjusted for different accuracy and dispense speed. Gear ratio can be adjusted to achieve different accuracy and speed as well. The biggest theoretical concern is the issue with jamming. Since spices can have non-trivial crystal size, a crystal might be able to jam the gumball wheel.

During testing, we found no major drawbacks. The jamming didn't occur while we tested the component. The amount of frictional force was not too large, making it possible for our DC motor to handle.

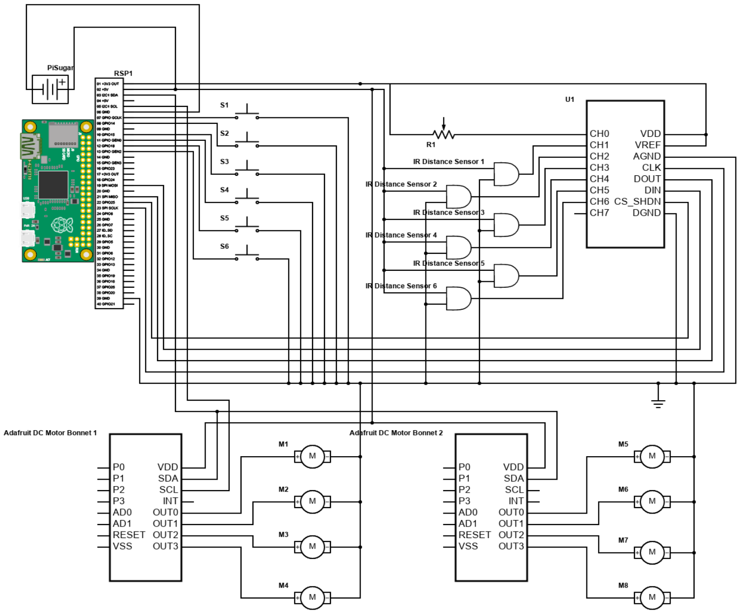


Fig. 2. Electronic Design, larger version at end of document on page 9

V. System Description

Our system's design is most heavily influenced by the idea that our project will be used as an everyday kitchen appliance, so we aim to have the electronics take as little space as possible within the device, yet still capable of performing all necessary functions in real time. With this goal in mind, we decided to use a Raspberry Pi Zero W because of its compact size, computing capabilities, various communication protocol interfaces, and wireless communication for our potential app (after MVP). Along with the Raspberry Pi Zero W, we utilize a PiSugar module to power the Raspberry Pi and the rest of the system. We picked the PiSugar because it consists of a battery controller, a rechargeable battery and it is designed to compactly mount on the bottom of the raspberry pi.

A. Actuators

The raspberry pi will control all the actuators of our system. Our design includes the use of a total of seven DC motors; we have six 6 mm DC 3V Precision Mini Micro Planetary Reduction Reduce Gear 4 Stage motors, one for each spice compartment and one MakerFocus Raspberry Pi DC Brushless Cooling Fan to help blow the spices rather than to cool the actual raspberry pi. The Raspberry Pi Zero W cannot drive enough power to control all seven DC motors, so we utilize two of Adafruit's DC Motor Bonnets designed to stack on top of the raspberry p. Thes motor bonnets can also be stacked on top of one another for more compact hardware. These motor bonnets are powered by the PiSugar module.

Each motor bonnet is capable of controlling four DC motors and can communicate with the raspberry pi through the I2C communication protocol interface. The I2C interface on the raspberry pi is pins 3 and 5; pin 3 is the data line (SDA) and pin 5 is the clock line (SCL). The raspberry pi can control all seven motors using only these two pins because the I2C protocol allows for 128 devices to communicate with the host. The address of each motor is offset by the address set on the bottom of the motor bonnet by soldering in the 7 bit address offset. So each motor will be driven by one of these motor bonnets and have a unique address to the raspberry pi

Since these motor bonnets are designed to be stacked on top of the raspberry pi, there is no additional circuitry required for connecting the interface, besides using 2x20 stack headers to connect the header pins. Each motor will sit within the bottom of the device, attached to a gear to spin a turnstile that causes the spice to dispense when on.

So in the control stage housing, we have a raspberry pi zero W mounted on top of a PiSugar module, and two motor controller bonnets stacked on top of each other mounted on top of the raspberry pi as seen in CAD model in figure B.

B. Sensors

The raspberry pi will also receive data from all the sensors of the system. The sensors we use are the six switches used for the dispense of each spice, six infrared distance sensors for measuring the amount of spice remaining in each compartment, and a potentiometer to select how much to dispense.

We use tactile switches that operate around 20 mA. Each of the six switches will map to motor through the raspberry pi to dispense a specific spice. They are simply connected to a GPIO pin of the raspberry pi and to ground. The raspberry pi will read if the button has been pushed and then tell the specific motor to turn on and dispense the spice of that compartment. These sensors do not require additional circuitry because we make use of the internal pull-up resistors for each GPIO port. The GPIO ports we use for each switch are as follows: GPIO 4 (pin 7), GPIO 14 (pin 8), GPIO 15(pin 10), GPIO 17 (pin 11), GPIO 18 (pin 12), and GPIO 27 (pin 13). These connections are also depicted in figure H and table H.

The IR distance sensors and potentiometer provides analog outputs, while the raspberry pi is only capable of digital inputs, so we also utilize an analog to digital converter (ADC MCP3008). This ADC has eight channels and communicates to the raspberry pi through the SPI communication protocol interface, so all of the analog sensors can be read by the raspberry pi through the one ADC.

	pin name	pin number
switch 1	GPIO 4	7
switch 2	GPIO 14	8
switch 3	GPIO 15	10
switch 4	GPIO 17	11
switch 5	GPIO 18	12
switch 6	GPIO 27	13

ADC 9	GND	9
ADC 10	GPIO 22	16
ADC 11	GPIO 10 (MOSI)	19
ADC 12	GPIO 9 (MISO)	21
ADC 13	GPIO 11 (SCLK)	23
ADC 14	GND	9
ADC 15	3.3V	1
ADC16	3.3V	1

Table H. Raspberry pi pin connections within the PCB

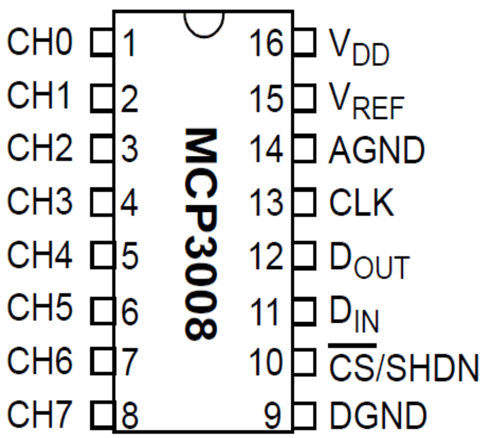


Fig. H The ADC we use and all it pins for reference in Table H [2]

The six IR distance sensors we use are the Sharp GP2Y0A51SK0F analog distance sensors which have a range from 2 - 15 cm. This range works well for our device because the height of each compartment is within that range. To determine the volume remaining for a spice, we would read the measured distance from the sensor and use the raspberry pi to map the measured distance to a specific volume. This works because the volume overall volume of the compartment is fixed. The IR sensor has three leads for power, ground and Vcc for the analog measurement. The six IR distance sensors are powered by the PiSugar's 5V port because these sensors operate between 4.5 and 5.5V. The Vcc value will go to one of the channels of the ADC and then sent to the raspberry pi to map to a specific distance, which would then map to a specific volume.

The potentiometer we use is the 12 mm Rotary position sensor. The rotary device will be connected to a rotating ring around the device through gears so it does not need to be exactly centered within the device. The ring around the device can be used to specify a specific measurement to be dispensed when the button is pushed. The potentiometer also has three leads for power, ground and Vref, which is the analog voltage that gets passed to the ADC and then sent to the raspberry pi where it can map the measured voltage to a position which is mapped to a certain amount to dispense. The amount to dispense dictates how long or fast to run the motor when it's button is pressed.

To minimize the circuitry for all of these sensors, we utilize a custom PCB that routes all the wires to the proper pins of the ADC and raspberry pi. All of the sensors will be wired into the custom PCB because the physical locations of these sensors are needed elsewhere in the device. The schematic for all the sensors routed by the PCB can be found in the schematic below.

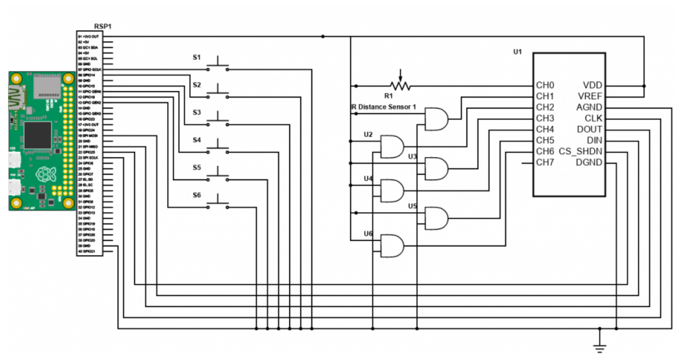


Fig. I PCB/Sensor Schematic

C. Software System

To tie all of these components together to perform the function of our device, we developed a software program on the raspberry pi to read from the sensors and determine what to do from those read values. Our program is event driven for controlling the dispense mechanism because our system only needs to be on when the user pushes the button. The raspberry pi will also only read from the IR distance sensors whenever the button is pushed or when the device is turned on because the volume of the spice is not expected to decrease unless the button is pushed.

Our program must initiate communication protocols for the SPI and I2C ports. We use SPI and I2C libraries for easier use and cleaner code when communicating to and from components via SPI or I2C.

When one of the six buttons are pressed, the raspberry pi will read the position of the rotary potentiometer through the SPI interface from the ADC, and then depending on the value read, the raspberry pi will communicate via I2C with motor controller to turn on the motor for a specific amount of time. The raspberry pi will know which motor to turn on by the port of the button that is pressed.

We implement a user friendly interface through a small HDMI display that can be driven by the HDMI port of the raspberry pi. This display will show the volume remaining for each spice.

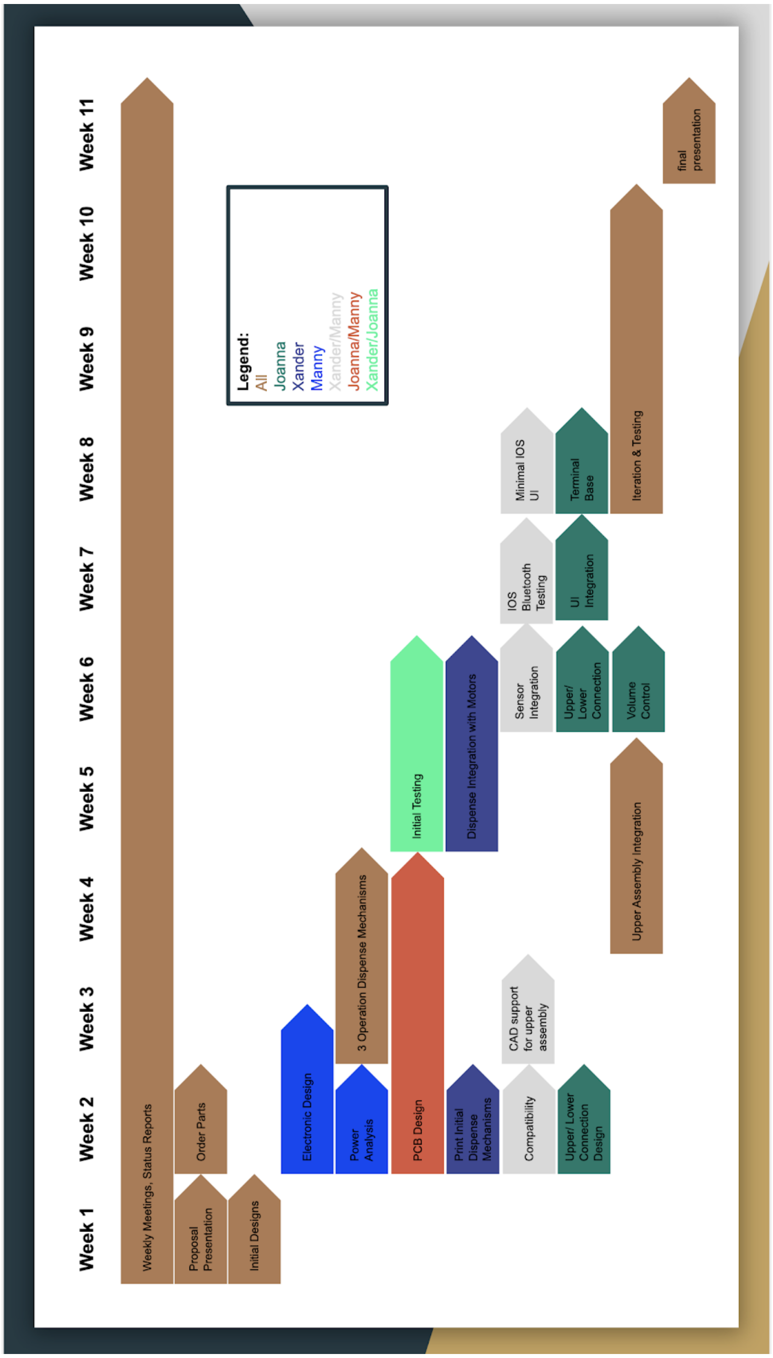


Fig. 3. Gantt Chart

VI. Project Management

A. Schedule

Our schedule remains steady, as shown in Fig. 3. Our modular design allows for us to work on components separately and will allow for easy integration. Many components on the plan are collaborative to ensure constant communication and to make integration easier. We have set out a lot of time for testing throughout the process. There were a few minor changes with timing due to waiting for parts and our whole team needing to be in Pittsburgh.

B. Team Member Responsibilities

Xander will be responsible for the mechanical design and implementation, as well as component interacting and IOS software design. Manny is responsible for the motor controller and interface, as well as PCB design. Xander and Manny will work together on integration and a minimal IOS UI design. Joanna will work with Xander on the interfacing as well as with Manny on the PCB design. She will also focus on internal project management, user input, and testing.

C. Budget

The budget can be found at the end of the document, on page 11. We have just exceeded half of our budget and have a remaining $170.76, leaving wiggle room for potential setbacks or design changes in the future.

D. Risk Management

Our main approach for risk mitigation was to test early and often and have legitimate fallback plans. Our schedule was borderline pessimistic, allowing us to have plenty of wiggle room. We also have been conservative with our budget and expect to have funds leftover for unexpected changes. Each of our validation metrics have unique fallback plans. For weight, we will simply 3D print a shell with thinner walls, playing at tradeoffs between function and form. The width of the design is stricter, but we have wiggle room with the height. Our battery life estimates were extremely conservative, so if there is any issue here we might need to reassess our requirement or look to see where in the product we are struggling. To address dispense accuracy, we have three kinds of mechanisms being tested in parallel. For dispense speed, we have left the option to adjust the gear ratio. The spray nozzle is flexible and modular so we can easily redesign and replace to adjust the spray ratio. If the noise level is too high, we can make the 3D printed parts more robust, which is the same solution laid out for the jamming rate. Thus far we have not faced any of our potential setbacks and are on course for MVP.

VII. Related Work

Our solution is very unique in its market, i.e. there are no major "competitors". Many spice racks exist, but not in the portable *and* automated way we are able to accommodate. Spice racks typically just hold individual shakers and scales/ measuring cups don't have a storage compartment. The closest device to what we have is the "KitchenArt Select-A-Spice Auto-Measure rack" [1]. It stores and measures spices but does not offer portability.

VIII. Summary

Thus far our design is on track to reach our specifications for MVP. Initial testing of the different dispense mechanism as shown promise for a successful final design. We are limited for MVP to having the device function for 1-2 spices instead of its capacity for 5-6, which technically limits the performance. We will be able to elaborate further on the system performance as we reach the integration stage.

A. Future work

We are still considering whether or not to continue this work after the semester ends. We are confident in our use case and believe this product would be incredibly useful for a large consumer group.

B. Lessons Learned

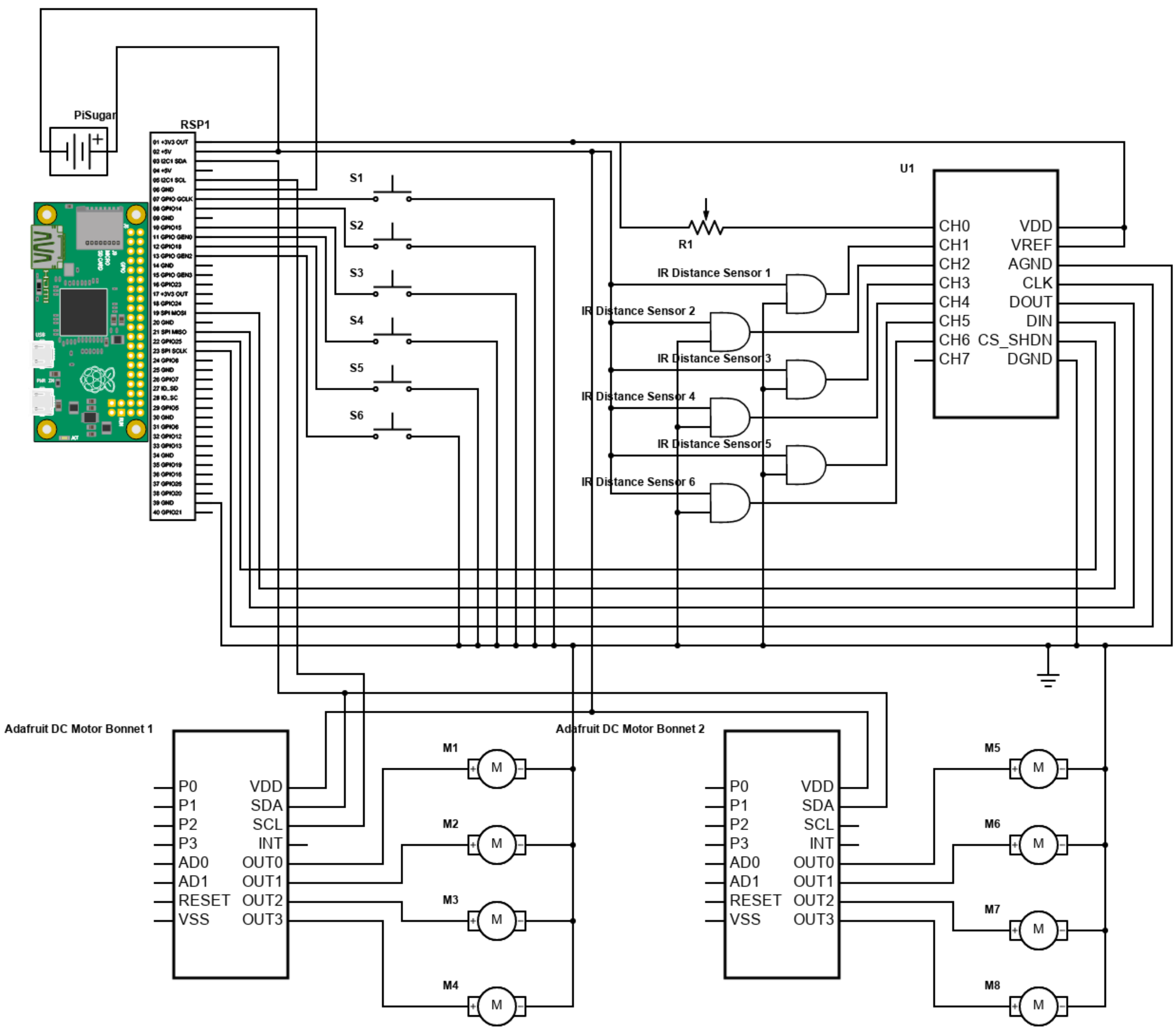
Our main focuses, and recommendations to others, is to focus on modularity of design and to integrate as soon as possible.

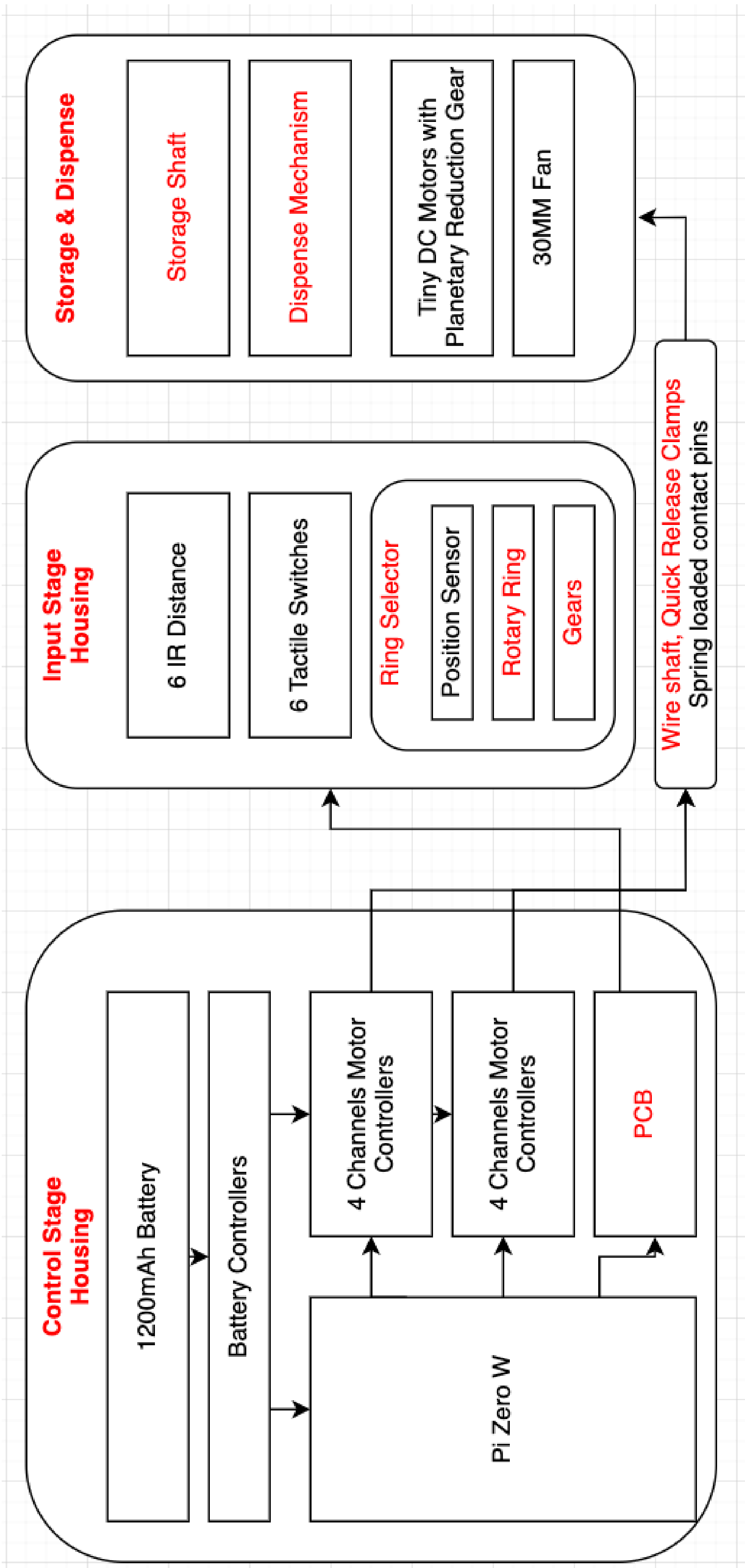
References

[1]https://www.amazon.com/KitchenArt-Select-Auto-Measure-Carousel-Professional/dp/B0000VLQTS

[2] Sklar, Michael. "Analog Inputs for Raspberry Pi Using the MCP3008." Adafruit Learning System, 12 July 2012, learn.adafruit.com/reading-a-analog-in-and-controlling-audio-volume-with-the-raspberry-pi/connecting-the-cobbler-to-a-mcp3008.

[3] https://www.iacacoustics.com/blog-full/comparative-examples-of-noise-levels.html


PiSugar
RSP1
S1
S2
S3
S4
S5
S6
U1
R1
CH0
CH1
CH2
CH3
CH4
CH5
CH6
CH7
VDD
VREF
AGND
CLK
DOUT
DIN
CS_SHDN
DGND
IR Distance Sensor 1
IR Distance Sensor 2
IR Distance Sensor 3
IR Distance Sensor 4
IR Distance Sensor 5
IR Distance Sensor 6
Adafruit DC Motor Bonnet 1
Adafruit DC Motor Bonnet 2
P0
P1
P2
P3
AD0
AD1
RESET
VSS
VDD
SDA
SCL
INT
OUT0
OUT1
OUT2
OUT3
M1
M2
M3
M4
M5
M6
M7
M8
M


Storage & Dispense
Storage Shaft
Dispense Mechanism
Tiny DC Motors with Planetary Reduction Gear
30MM Fan
Input Stage Housing
6 IR Distance
6 Tactile Switches
Ring Selector
Position Sensor
Rotary Ring
Gears
Wire shaft, Quick Release Clamps
Spring loaded contact pins
Control Stage Housing
1200mAh Battery
Battery Controllers
4 Channels Motor Controllers
4 Channels Motor Controllers
PCB
Pi Zero W

Item	Price per Unit	Quantity	Total Cost
Raspberry pi zero W	$27.99	1	$27.99
Pisugar Portable 900 mAh /1200 mAh Lithium Battery Power Module for Raspberry Pi-Zero	$39.99	1	$39.99
6mm DC3V Precision Mini Micro Planetary Reduction Reducer Gear 4 Stage	$12.49	1	$12.49
Tactile Switches, Round Actuator Style, Operating Function NO, Copper Alloy, Silver, Low Profile, 20mA Tactile Switch Contact Current Rating	$0.80	6	$4.80
mxuteuk White Aluminum Alloy + Black Rubber Ring Potentiometer Control Knob	$9.26	1	$9.26
Mxfans 100x Golden Plating Copper Spring Pogo Pins Probes 2mm Dia 6mm Height	$11.52	1	$11.52
Adafruit DC & Stepper Motor Bonnet for Raspberry Pi	$22.50	0	$0.00
Adafruit DRV8833 DC/Stepper Motor Driver Breakout Board	$4.95	4	$19.80
Rotary Pot	$2.59	2	$12.17
MCP3008 - 8-Channel 10-Bit ADC With SPI Interface	$3.75	2	$18.93
Stacking header	$1.95	1	$1.95
Extra long breakaway headers	$9.59	1	$9.59
3M 3313 Copper Foil Tape - 1/2" x 18 yds	$11.49	1	$11.49
ABS resin (Translucent, 1000g)	$36.99	1	$36.99
MakerFocus 2pcs Raspberry Pi DC Brushless Cooling Fan	$5.49	1	$5.49
Sharp GP2Y0A51SK0F Analog Distance Sensor 2-15cm	$11.23	6	$77.33
Extra Tall Stacking Header	$2.95	1	$2.95
Ribbon Cable	$2.95	1	$2.95
Professional Ultra SanDisk 32GB MicroSDHC Raspberry Pi 3 Model B+ card	$11.99	1	$11.99