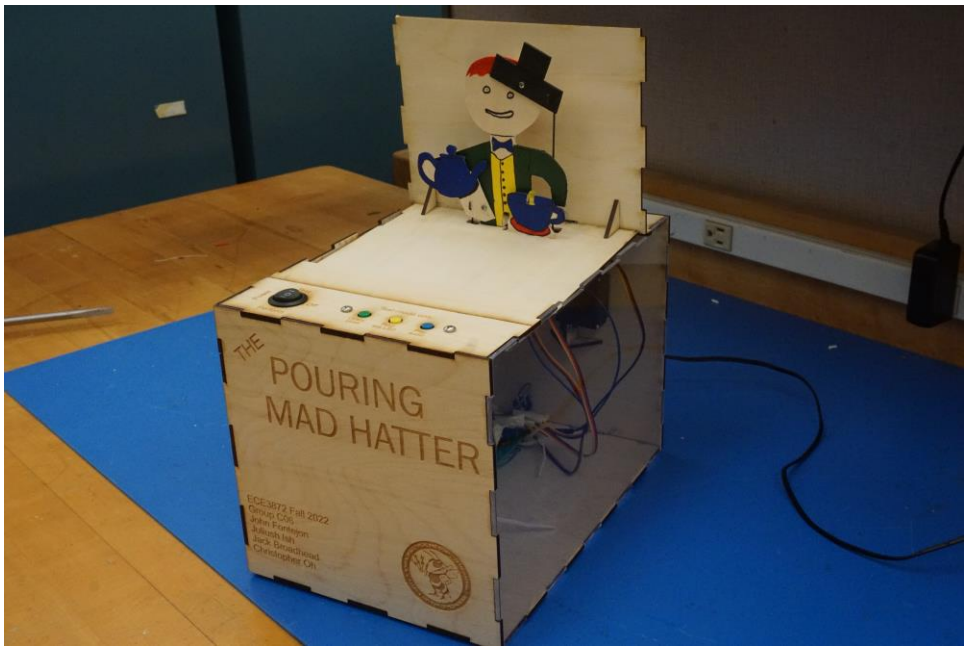
Georgia Tech

CREATING THE NEXT



Design Document
The Pouring Mad Hatter

Team #C06: Jack Broadhead, John Fontejon, Julius Ish, Christopher Oh

Date: October 23, 2022

Table of Contents

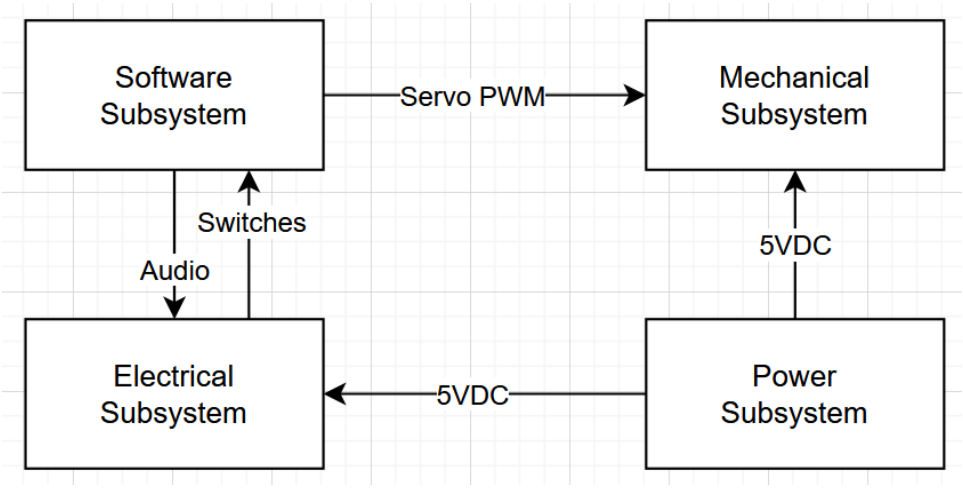
Revision Record

Date	Author	Comments
September 20, 2022	Team	Document Created (framework)
September 22, 2022	Jack John	Added mechanical details, schedule, BOM Added electrical details
September 23, 2022	Chris	Added power system details, project description
September 24, 2022	Julius	Added software system details, software architecture
September 25, 2022	Team	Added team details
October 1, 2022	Chris	Added details to power subsystem
October 19, 2022	Chris John	Added details to power subsystem Added details to electrical subsystem
October 21, 2022	Jack	Added mechanical subsystem info, updated BOM
October 23, 2022	Team	Added details for CDR
December 6, 2022	Jack John	Updated mechanical subsystem info (incl. pictures) Updated electrical subsystem info (parts table)

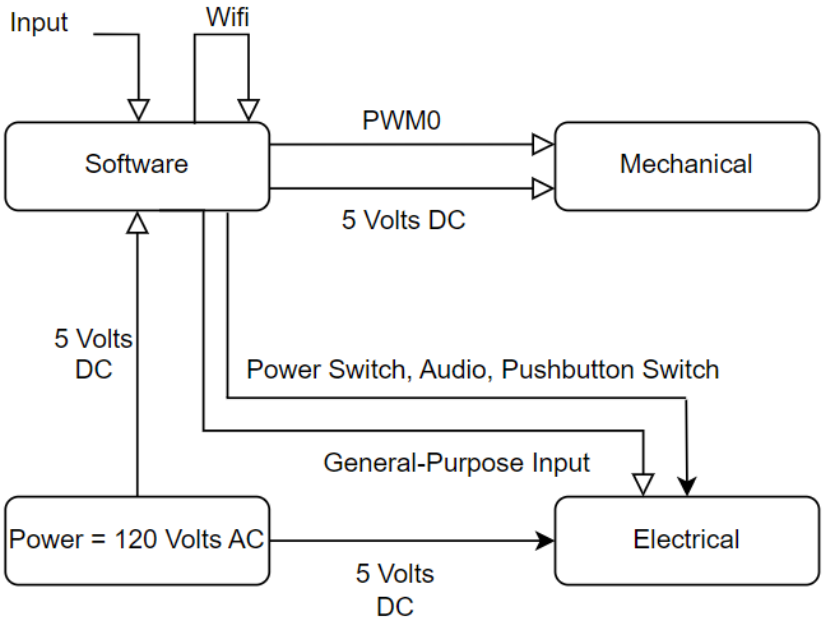
Project Description

This project tasks our group with creating a robot that fits into the "Alice of Wonderland" genre that will be able to recite theatrical lines that are sent in from a director. The robot has size constraints which will be discussed later along with basic actions such as growing/shrinking. For our robot, we have decided to replicate the famous "Mad Hatter" character pouring tea into a cup, while an LED in the cup will glow and a speaker will create a "pouring" sound. To ensure our design fits criteria and work is done in timely fashion, the team has assigned subsystems to each individual based on their strengths and preferences.

System Design

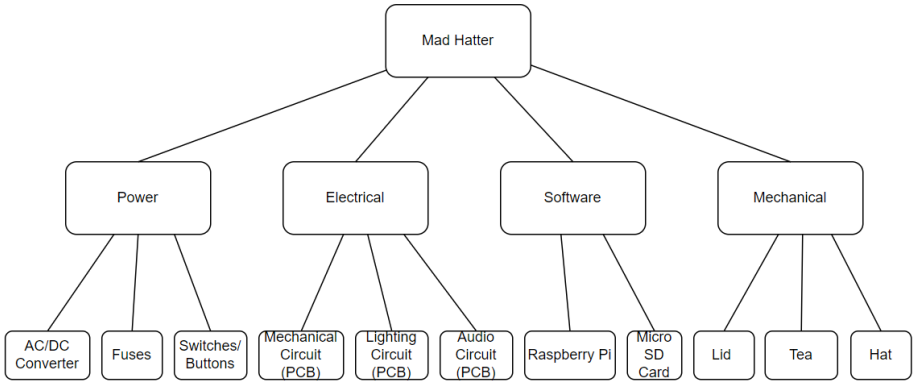


This figure shows each subsystem and how it interfaces with the others. Here is a closer look at the subsystems and how they are working with each other.



Sub-System Design

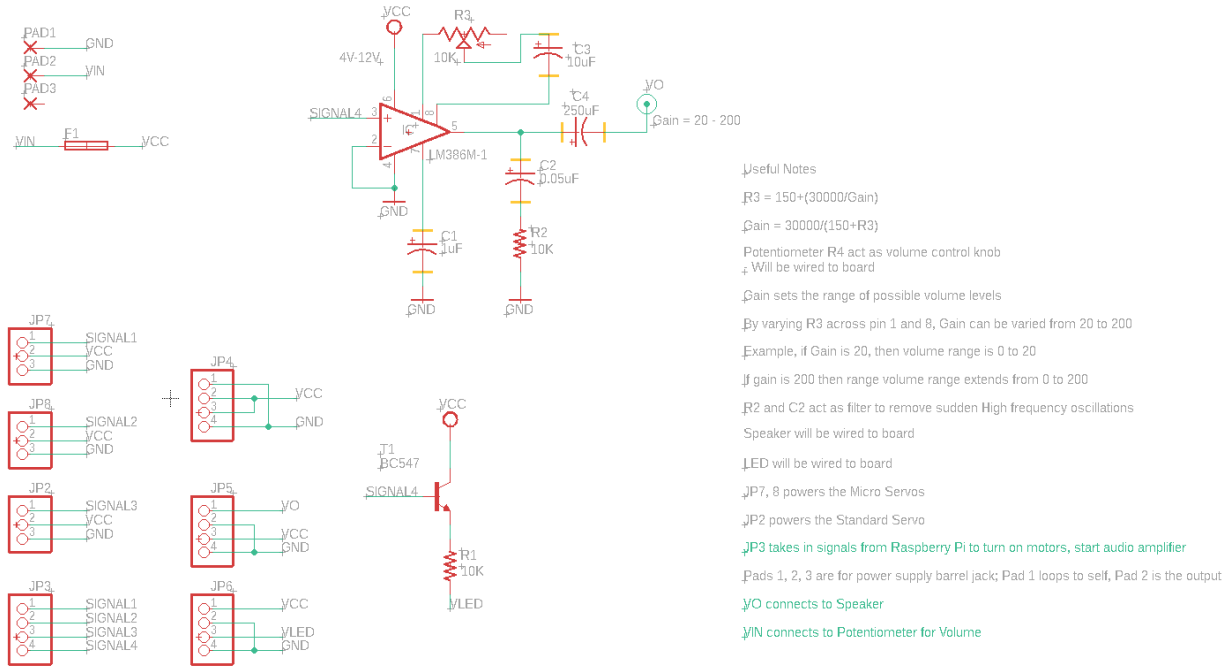
With the customer needs and a general idea of our overall design, we split our system into four systems that will cohesively fit together at the finality of our project that will result in the robot. The subsystems are electrical, mechanical, power, and software. Electrical encompasses the PCB design along with various other connections to be discussed soon. Mechanical involves motor placements, creating prototypes of the design, and more. The power subsystem will draw in power from a wall plug and will power the whole box and has a switch for the user input. The software will be able to connect to a director.



Subsystem Leads

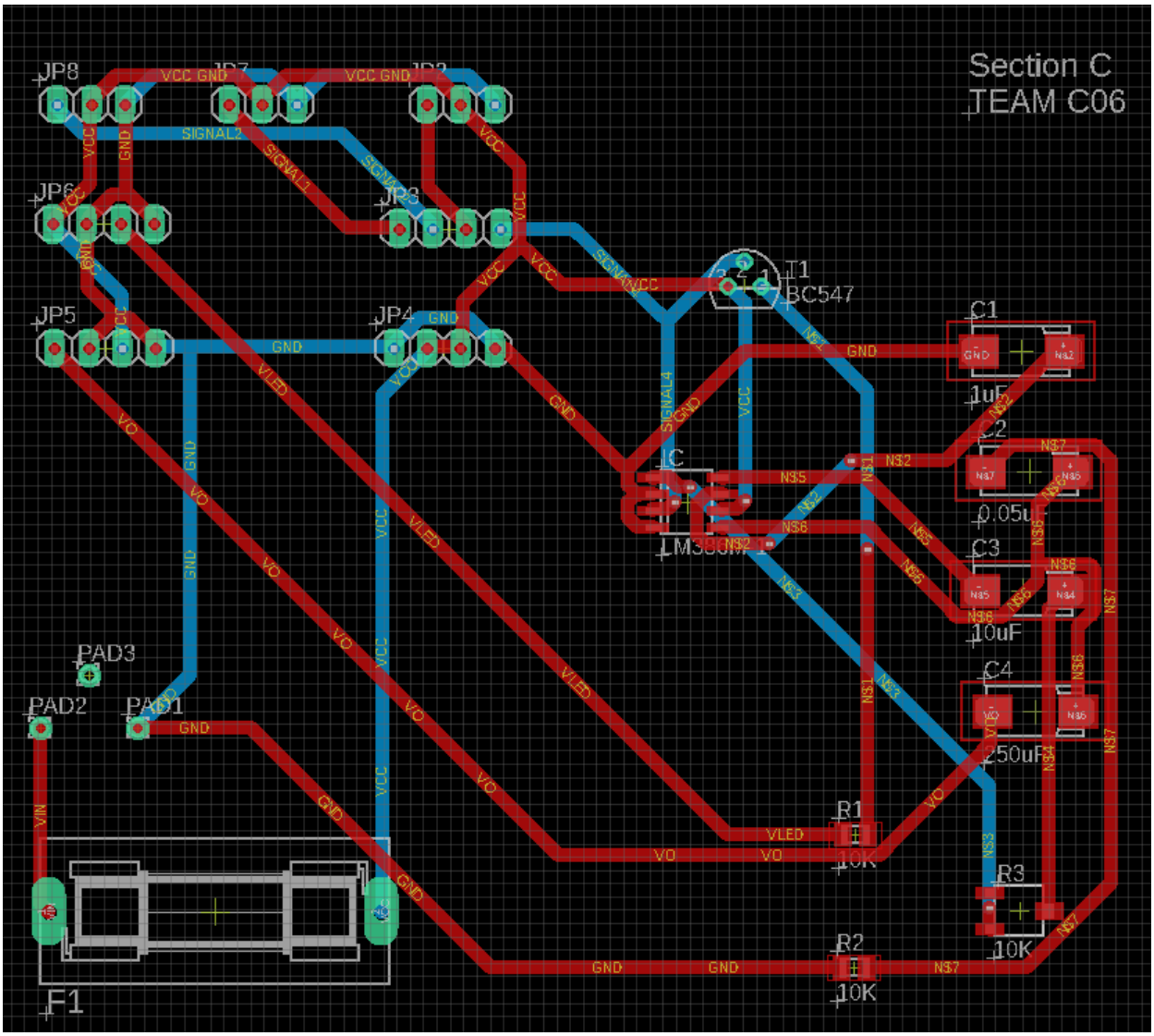
Power – Chris Oh (Team Lead)
Electrical – John Fontejon
Mechanical – Jack Broadhead
Software – Julius Ish

Electrical Subsystem

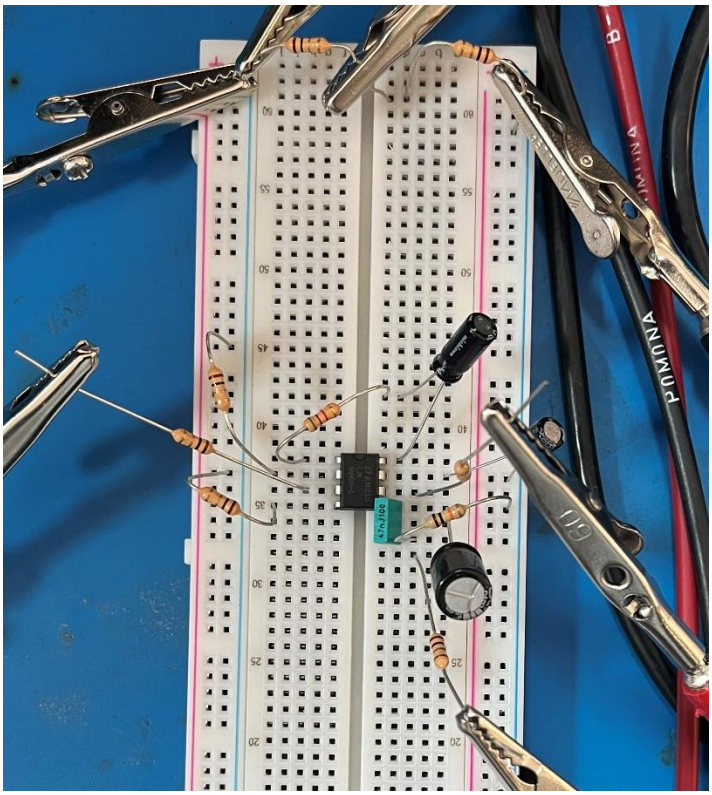


This subsystem will consist of a lighting circuit and an audio circuit, as well as pin headers for external connections. The input power will go through an AC/DC converter and into a 2.5 A fuse before providing power to the entire PCB. The audio circuit, pictured above, will include an op-amp to increase the output and a speaker to produce noise. A low pass filter is used to reduce the noise output of the op amp. There will be multiple pin headers to provide power to the servos, LED, and Raspberry Pi and to receive signals from the Raspberry Pi to activate those sub-systems. A list of all the components used can be found in the following table.

Component	**Value(s)**	**Quantity**
Resistor	10 kΩ	2
Potentiometer	10 kΩ	1
Capacitor	0.05 uF, 1 uF, 10 uF, 250 uF	1 each
Fuse	2.5 A	1
Fuse Holder	N/A	1
3 Pin Header	N/A	3
4 Pin Header	N/A	4
LM386-N Op Amp	N/A	1
BC547 NPN Transistor	N/A	1
Barrel Jack	N/A	1
Speaker	N/A	1

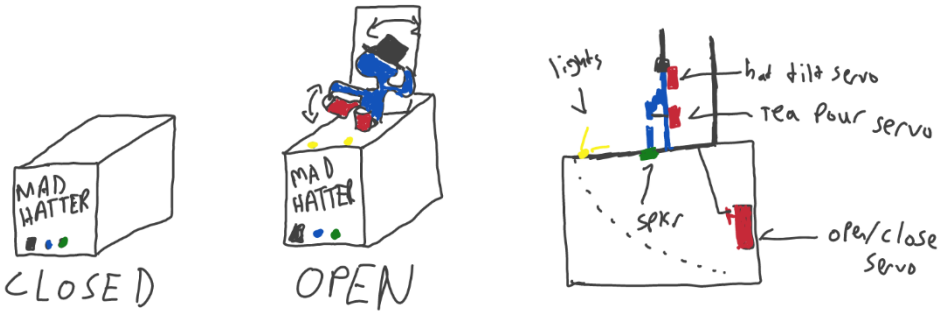


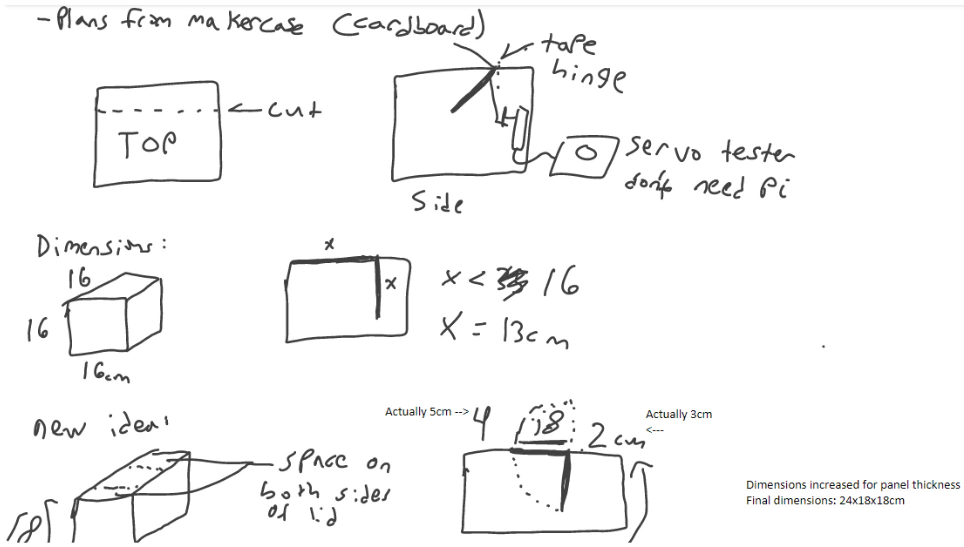
The board has two layers, with every component placed on the top layer.



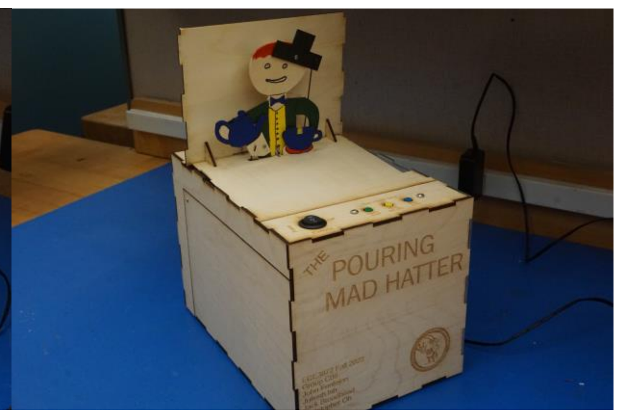
Testing the designed audio circuit, which is comprised of an op-amp with an adjustable gain and a low-pass filter on the output. This testing circuit was used for the electrical simulation.

Mechanical Subsystem




- Plans from makercase (cardboard)
cut
Top
tape hinge
Side
servo tester
don't need Pi
Dimensions:
16
16
16cm
x
x
x = 13cm
new idea!
space on both sides of lid
Actually 5cm -->
Actually 3cm
<---
2 cm
Dimensions increased for panel thickness
Final dimensions: 24x18x18cm


THE
POURING
MAD HATTER

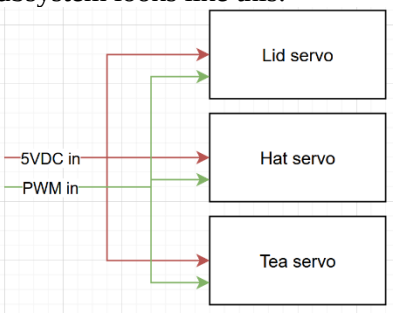

THE
POURING
MAD HATTER

The subsystem consists of the box structure to house our robot and all the movement mechanisms. The box is made of laser cut plywood, with a special lid that swings up, revealing a stage with the Hatter himself inside. When powered off, the robot appears as a simple wooden box. However, when prompted, a servo rotates the lid open, revealing the robot. The Hatter is made of thin wood, with additional pieces for the hat, tea kettle, and hat. These pieces give the robot a layered, 3D appearance with the simplicity of a few small pieces of wood. A yellow LED is glued to the top of the tea cup. Two micro servos rotate the hat (giving the illusion of waving it) and the tea kettle (giving the illusion of pouring tea). Each servo is mounted to the bottom side of the robot, so they remain hidden inside the box. They connect to the hat and kettle using pushrods. The hat and kettle each rotate around a small bolt and locknut holding them to the body of the Hatter.

The box also houses the circuit board and raspberry pi, mounted with standoffs. All wiring is terminated with standard female pinheader terminals (or spade connectors for the power switch), allowing easy connection and disconnection. A control panel-with power switch and buttons to trigger each action- is mounted on top of the box. There is plenty of space inside to house all the components, with one side being removable for internal access, and the other being clear plexiglass for a peek inside.

The subsystem receives PWM signals from the software to control the position of each servo, and the servos are powered by 5VDC by the power subsystem. Plus, the LED mounted in the teacup is powered by the Raspberry Pi.

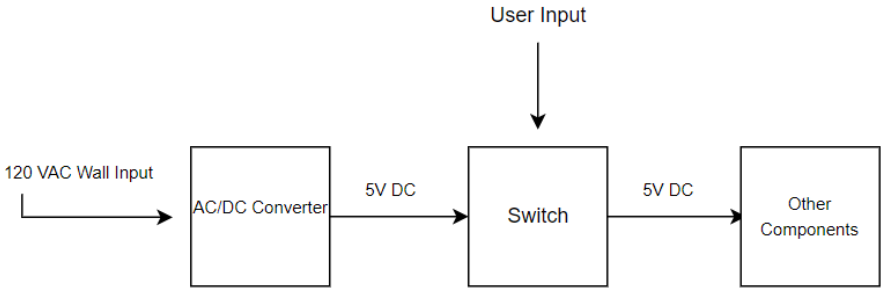
On a component level, the subsystem looks like this:



Power Subsystem

The power subsystem starts with a 120 VAC to 5 V DC converter. This wall plug power source will be connected to the PCB our team builds. The PCB will have a 5V and a GND rail that connects to the Raspberry Pi and since the source has a fuse built into it, we do not need to worry about overpowering the Pi. However, as a precaution we will have a fuse on the PCB as well that will fit under the 5V and 2A provided by the supply. The switch we have has a neutral option along with two other outputs. The middle prong will be connected to the voltage source (5V) and the other prongs will represent the final mode of our robot during demo stage, and the other will be for the test mode to ensure our robot behaves

as intended. The power source will then power the Raspberry Pi, along with the other components that require external power, such as the speaker, etc.



Next to be shown is a table that shows where all the power is going.

Source	Destination	Description
System Input - 120VAC	Power Subsystem	Wall wart powering the system
Power Subsystem - 5VDC	Servos, speaker, light, Raspberry Pi	5VDC power for the servos, the speaker, the light inside the cup, and the microprocessor
Raspberry Pi – Digital signal	Speaker, lights, servos	The digital signal goes to the speaker, lights, and servos
System input - Network	Raspberry Pi	Connection to director commands

The power supply we chose is a 5V and 2A wall wart that can be connected to a barrel jack. It is 120VAC that is converted into 5VDC which powers many other systems in our design for the robot. This meets the necessary requirement given by the customer, which is the power must be less than a certain wattage (60 Watts). The major question remains, which is would the power system be able to power the PCB, the Raspberry Pi, along with many other components. To find if this is a reasonable power supply, Christopher went into a lab with parts to see if the power budget would fit under the 5V and 2A of the supply we were using. After testing, the values were put in the table below.

Part to be used	Voltage	Amperage
Big servo motor	5.0V	300mA
Small servo motor	5.0V	150mA
Speaker	5.0V	0.11A
Raspberry Pi	5.0V	0.13A
Total	5.0V	0.69A

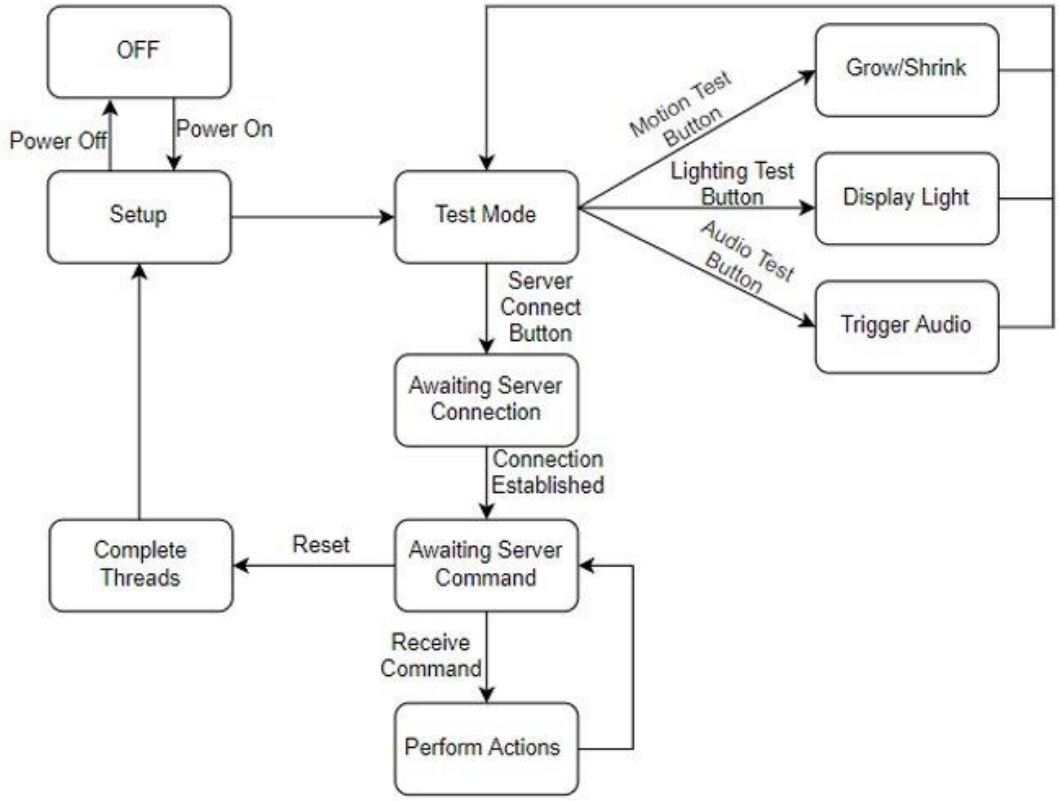
Based on the table, we can see that the voltage makes sense. Even better, is that the total amperage is under the 2 A requirement, so we can use the power supply for our design. Also, it should be noted that for the amperage column, Chris took the worst-case scenario for the measurements in the lab. These components were each hooked up one at a time, and even when all were connected, the amperage was much lower than 2A. To ensure that there will be no burnouts, we will still use fuses just to triple guard our circuit from being blown up. On initial discovery of this, Chris and John made multiple

communications on how to move forward with the PCB design, and the thought is to have a fuse holder so that even if a fuse is blown the replacement can be quick and easy.

The PCB takes in 5 V and 2 A from the power supply. The Watts provided is equal to the voltage times the current. This makes the output 10 Watts. This is important because in order to find the fuse we need, we need to find the amperage + 10% of the amperage. In our case, that is 2.2 Amps, so we need a fuse that is at least that value. The closest value that is at least that is the 2.5 A fuse.

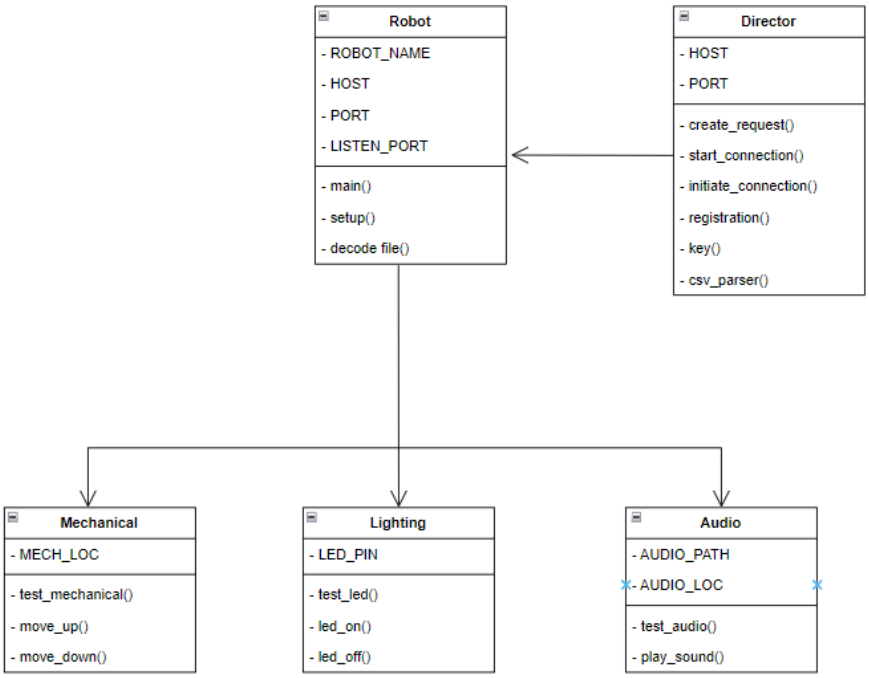
Software Subsystem

The software subsystem includes a Raspberry Pi 4 and tools to interface with other subsystems. The raspberry pi takes in a 5V power source and receives instructions from a remote server which prompts it to send out GPIO, PWM, and analog audio signals accordingly. Below is a state diagram detailing the various states possible between the software subsystem being turned on and sending signals to perform the Mad Hatter's core functions. To assist in debugging and ensure that the hatter can demonstrate its functionalities in the event of failure to connect to the server, the software subsystem includes a test mode in which each core functionality can be tested separately via input from the device user.



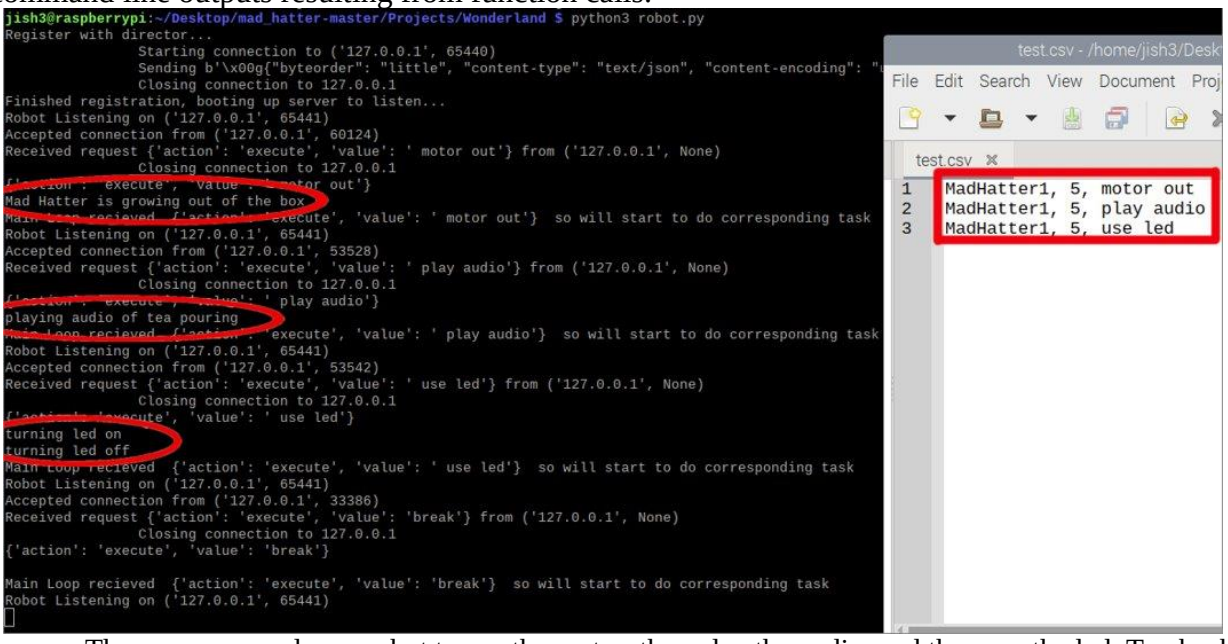
Software Architecture

Below is the preliminary software architecture detailing the classes, variables, and methods used. Each functionality, mechanical motion, lighting, and audio, has its own class with methods and variables necessary for their independent functioning. The robot class holds the main function, meaning that objects and methods defined in other classes will be invoked within robot's main method. The robot class also most directly interfaces with the raspberry pi and the rest of the Mad Hatter robot hence its name. The director class includes the functionality to connect with the server and receive instruction, which are then handled by the robot class.



Software Simulation

To test the basic architecture of our software, a simulation was run that involved connecting to the director and receiving commands from a prewritten csv file to trigger the calling of appropriate functions. Below is a screenshot displaying the csv file contents as well as the command line outputs resulting from function calls.



The csv commands our robot to use the motor, then play the audio, and then use the led. To check that our robot receives and processes these commands correctly, we can verify that the three circled command line outputs are printed in the correct order, with the motor associated output first, then the

audio, and finally the leds. With this simulation complete, all further developments in the software involve interfacing with hardware in the electric and mechanical subsystems.

Test Plan

As will be indicated on the test matrix to follow, our tests measure how well our design fits the customer requirements along with functionalities that we want to add on. The first tests involve basic restraints for the size of our robot, along with the most necessary functionality required by the employer.

System Decomposition

Number	Customer Need	System Requirement Number	System Requirement	Test Type	Test Number
1	The robot must receive and recite lines from a server through a CSV file script	Sys_1.1	Robot will have wifi connection with server	T	6
2	The robot must have some sort of lighting, have at least one motion in which it grows or shrinks, and produce sound during line delivery	Sys_1.2	Robot will have a speaker to recite lines	I	1
3	The robot must have one analog circuit (PCB)	Sys_1.3	Robot will have a software architecture to decode CSV file	D	4
4	The robot must have a power source with a panel to switch on/off	Sys_1.4	Robot will recite lines from a decoded CSV file when prompted by the server	T	6
5	The robot must have a test mode	Sys_2.1	Robot must fit within a 35cm x 35cm x 35cm box constraint when turned off	I	2
		Sys_2.2	Robot must fit within a 75cm x 75cm x 75cm box constraint when turned on	I	2
		Sys_2.3	Robot will use servos to increase height and "grow"	D	3
		Sys_2.4	Robot will have a speaker to recite lines as the servos are activated	D	4
		Sys_2.5	The robot will have the cup light up as the "tea" is poured in	D	3
		Sys_3.1	PCB dimensions will fit within a 15cm x 15cm space and will be a 2-layer design	I	1
		Sys_3.2	Must be easy for user to access PCB to allow changes/fixes	I	1
		Sys_3.3	Robot's sound will be controlled by the analog circuit	D	3
		Sys_4.1	Must have a 120 VAC input	I	2
		Sys_4.2	Must consume less than 60 watts	T	5
		Sys_4.3	Must have an AC/DC converter to transform the 120 VAC input to usable 5VDC for components	T	5
		Sys_4.4	Robot will have a switch that can be turned on by user to activate all the following processes to fulfill requirements	T	5
		Sys_4.5	Robot will have fuses to power all the other components in the robot such as audio and PCB	I	2
		Sys_5.1	The robot will have a switch to put the robot into test mode	I	2
		Sys_5.2	When a switch is flipped the robot will test audio and height features separately	D	3
		Sys_5.3	The software architecture will implement the test mode when switched	D	4

Click here for the full Test Plan

Test Matrix

	Requirement Number																			
Test Number	Sys_1.1	Sys_1.2	Sys_1.3	Sys_1.4	Sys_2.1	Sys_2.2	Sys_2.3	Sys_2.4	Sys_2.5	Sys_3.1	Sys_3.2	Sys_3.3	Sys_4.1	Sys_4.2	Sys_4.3	Sys_4.4	Sys_4.5	Sys_5.1	Sys_5.2	Sys_5.3
1		I								I	I									
2					I	I							I				I	I		
3							D		D			D							D	
4			D					D												D
5														T	T	T				
6	T			T																

Schedule

	Week Number												
Task	1	2	3	4	5	6	7	8	9	10	11	12	13
Brainstorm	Chris												
Proposal		DUE											
Block Diagram	Jack												
Software State Diagram	John												
PDR					DUE								
Design Document		Julius											
Test Plan		Chris											
Schedule	Jack												
BOM		John											
Notebook (Meeting Notes)	Chris	Jack	John	Julius		Jack	John	Julius	Chris	Jack	John		
CDR: Before we build							DUE						
Design Document				Julius									
Test Plan		Chris											
Schedule				Jack									
BOM				John									
Notebook (Meeting Notes)	Chris	Jack	John	Julius	Chris	Jack		Julius	Chris	Jack	John		
System Integration			Jack										
System Test					John								
Simulation Video						Julius							
Electrical Subsystem (John)													
Power Circuit			John		Jack								
Lighting Circuit			John		Jack								
Interface Circuit (Raspberry Pi)			John		Jack								
Audio Circuit				John		Jack							
Testing Circuit					John		Jack						
Mechanical Circuit					John		Jack						
Compile Circuits into 1 PCB							John	Jack					
Fabricate PCB								John					
Test								John			Jack		
Integration										John			
Documentation		John					Jack	John		Review (all)			
Power Subsystem (Chris)													
Power Conversion Rough Design			Chris										
Find Switch/Fuses/Converter			Chris										
Gather initial power budget					Chris								
Buy pushbuttons, fuse, etc.						Chris	Julius	Chris					
Connect with PCB									Chris				
Test								Chris			Julius		
Integration										Chris			
Documentation		Chris					Julius	Chris		Review (all)			
Software Subsystem (Julius)													
Finalize Softare States/Familiarization		Julius											
Wireless Cabilities			Julius										
Program Software States/Architecture			Julius										
Program Full Implementation				Julius									
Lighting Test					Julius								
Mechanical Test					Julius								
Audio Test					Julius								
Software System Test								Julius			Chris		
Integration										Julius			
Documentation		Julius					Chris	Julius		Review (all)			
Mechanical Subsystem (Jack)													
Rough design (figure scope)		Jack					John						
Box design				Jack			John						
Box lid mechanism prototyping						Jack	John						
Character design (including prototyping)					Jack		John						
Build box (includes 'growing' mechanics)							Jack						
->laser cut wood							Jack						
->install servos									Jack				
Build character								Jack					
->laser cut wood								Jack					
->paint									Jack				
Test movement									Jack		John		
Integration										Jack			
->install lights+speaker in box										Jack			
->install PCB											Jack/John		
->final software tweaks													
Documentation		Jack					John	Jack		Review (all)			
Final Inspection and Demonstration											John	DUE	

Bill of Materials (BOM)

Note: datasheet links are omitted for brevity, but are included in the full Excel file.

Part Name	Part Num	Prototype Quantity	Purchase location	Physical location	Order status	Cost Each	Production Quantity	Production Cost	Misc
Standard Servo	155	1	Parts shop	Jack	Picked up	$ 12.00	$ 1.00	$ 12.00	
Micro Servo	169	2	Parts shop	Jack	Picked up	$ 5.95	$ 2.00	$ 11.90	
Raspberry Pi Four	n/a	1	Parts shop	Julius	Picked up	$ 10.00	$ 1.00	$ 10.00	
Servo pushrod kit (5x package)	n/a	1	Amazon	Jack	Picked up	$ 10.38	$ 0.20	$ 2.08	
Hinge (2x package)	15161	1	Home depot	Jack	Picked up	$ 2.65	$ 1.00	$ 2.65	
Yellow LED	TL-4253	1	Parts shop	Jack	Picked up	$ 0.95	$ 1.00	$ 0.95	
120VAC to 5VDC converter	n/a	1	Parts shop	Chris	Picked up	$ 8.00	$ 1.00	$ 8.00	
Pushbuttons colored 4 pack	14460	1	Sparkfun	Chris	Picked up	$ 1.18	$ 1.00	$ 1.18	
Switch	360-2388-ND	1	Digikey	Chris	Picked up	$ 5.48	$ 1.00	$ 5.48	
64GB Micro SD Card	n/a	1	Parts shop	Julius	Picked Up	$ 9.85	$ 1.00	$ 9.85	
Jumper Wires	1528-1155-ND	2	Digikey	Not yet picked up	Not yet ordered	$ 3.95	$ 1.00	$ 3.95	
Speaker	11089	1	Sparkfun	Chris	Picked up	$ 2.10	$ 1.00	$ 2.10	
Barrel Jack	119	1	Sparkfun	Chris	Picked up	$ 1.50	$ 1.00	$ 1.50	
10K Resistors	Y402310K0000T9R	2	Digikey	Not yet picked up	Not yet ordered	$ 0.02	$ 2.00	$ 0.04	Actual cost only pennies each
Fuses	F2681-ND	5	Digikey	Not yet picked up	Not yet ordered	$ 0.53	$ 1.00	$ 0.53	
Trim Resistor	3223G-1-103E	1	Digikey	Not yet picked up	Not yet ordered	$ 3.88	$ 1.00	$ 3.88	
4 Pin Header	2015-1X04	4	Digikey	Not yet picked up	Have 2	$ 0.48	$ 4.00	$ 1.92	
Fuse Holder	SH25	1	Digikey	Not yet picked up	Ordered	$ 4.50	$ 1.00	$ 4.50	
Op Amp	LM386M	1	Digikey	Not yet picked up	Picked up	$ 1.30	$ 1.00	$ 1.30	
Transistor	BC550CBU	1	Digikey	Not yet picked up	Not yet ordered	$ 0.65	$ 1.00	$ 0.65	
3 Pin Header	2015-1X03	3	Digikey	Not yet picked up	Have 2	$ 0.23	$ 2.00	$ 0.46	
							Total:	$ 84.92	