Design Document

Queen of Hearts + Tarts

Team #C03: Geneve Lauby, Brooks Lewis, Lilly Sitver, Joshua Snyder

Date: November 28th, 2022

Table of Contents

Revision Record

Date	Author	Comments
Sep 13, 2022	Team	Document Created (system design and hierarchical design)
Sep 20, 2022	Team	Software Flowchart and Subsystem Descriptions updated
October 13, 2022	Team	Light Revision for LED Power Supply
October 20, 2022	Team	Updates to Subsystems and Simulation Results
October 23, 2022	Team	Updates to Integration and Simulations
November 7, 2022	Team	Updates to Subsystems
November 11, 2022	Team	Addition of Integration and Conclusion
November 19, 2022	Team	Updates to Software Subsystem
November 28th, 2022	Team	Addition of Project Photos and Edits of Subsystem

Task Leads

Team Member	Subsystem and Team Role
Geneve Lauby	Mechanical
Brooks Lewis	Audio, Light/Visual
Lilly Sitver	Project Lead, Software
Joshua Snyder	Power

Procedure

The robot requires both the USB and the wall plug-in to be plugged in to fully work. Next, flip the switch on the back of the box to turn the robot on. Once the robot is plugged in, the robot will take about 30 seconds to 1 minute to fully boot up. Once booted up, a user can press any of the pushbuttons on the robot's box for infinite play time.

Project Description

Our project is centered around the Queen of Hearts. This animatronic will sit on top of a wooden box that holds all electrical components. This box will control the communications to the audio, mechanical movements, and light visuals.

When the Queen of Hearts becomes angry, she will move her arm and the LED wand in hand will light up.

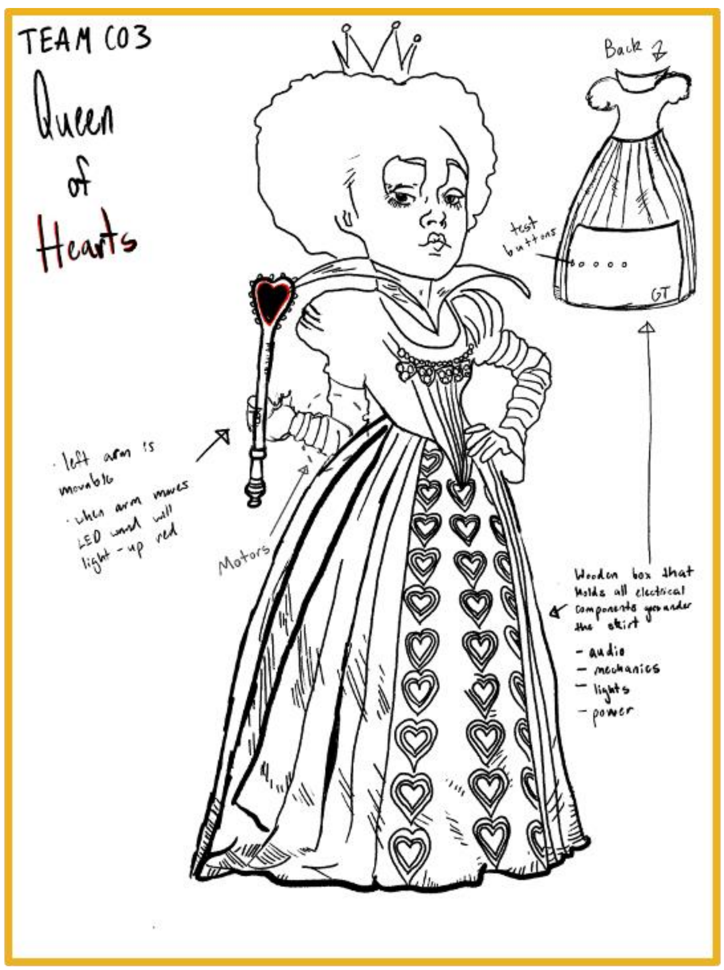

Figure 1: Initial sketch of the Queen of Hearts animatronic

Figure 2: Front of Final Queen of Hearts Animatronic

Figure 3: Back of Final Queen of Hearts Animatronic

System Design

The inputs of the system include 120V AC input power and user input. The outputs of this system include audio through a speaker, an LED light, and 180-degree arm movement driven by a servo motor. The system consists of five sub-systems including light and visual, software/processor, power, motion, and audio. The user interface includes an on/off button and test buttons for audio, light, movement, and test all. The processor is a Raspberry Pi Zero processor. The audio sub-system includes a 0.5W speaker and a custom analog audio amplifier. The mechanical system includes a DC servo motor for motion and custom laser cut arm. The power sub-system includes a 120V AC to 5V DC transformer, a user input switch, and a fuse. The software/processor subsystem includes an interface that connects the director, the Raspberry Pi, and the subsystem motors. The light/visual subsystem includes a red LED and a 3D printed wand.

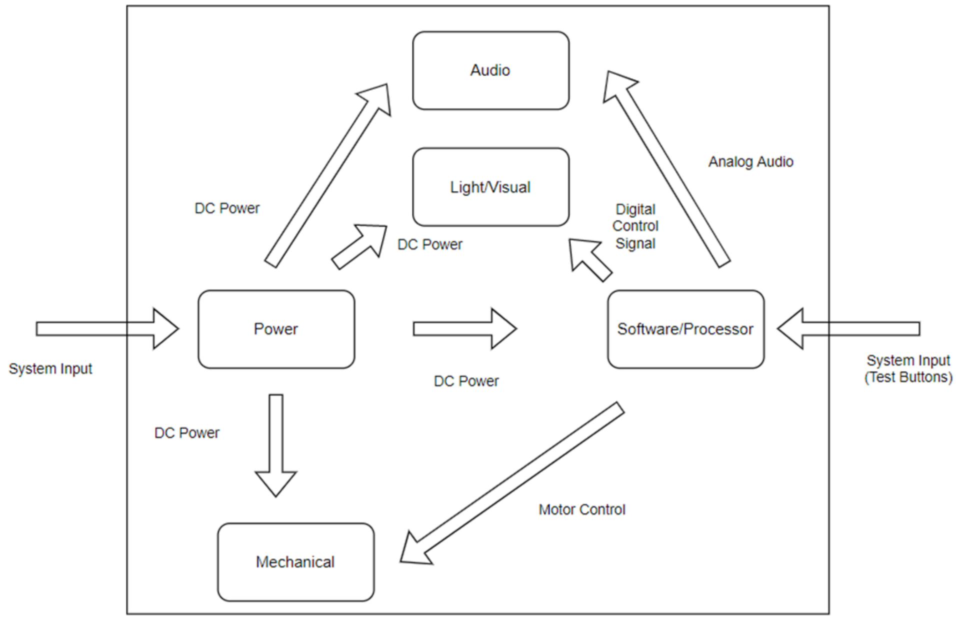

Figure 4: I/O System Diagram

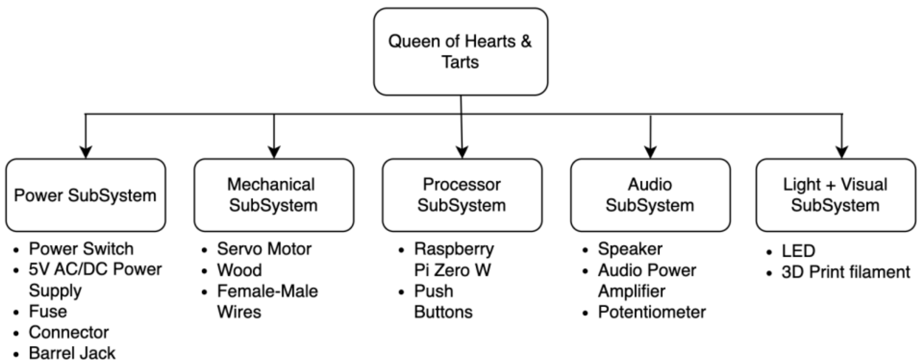

Figure 5: Block System, Subsystem, and Component Diagram

Interface	Source	Destination	Description
120 V AC	System Input	Power	120 V AC Input
5 V DC	Power	Processor	5V DC powers processor
		Audio	5V DC powers speaker
		Mechanical	5V DC powers arm movement
		Light/Visual	5V DC power LED light
User Input Pushbutton Controls	System Input	Processor	Digital input to processor.

Motor Controls Signals	Processor	Mechanical	Digital signal to control the mechanical arm.
Analog Audio Signal	Processor	Audio	Analog signal to output audio
LED Light Signal	Processor	Light/Visual	Digital signal to control the LED

Table 1: Subsystem Inputs and Outputs

Sub-System Designs

Our design incorporates five sub-systems: light/visual, software/processor, mechanical, power, and audio. The audio will process the raw tone signals and convert it to sound, mechanical will execute the arm movement of the figure, the LED will light up during arm movement, and the power will convert 120V AC to 5V DC. The software processor will integrate all the sub-systems by converting the user input to signals for the audio, light, and movement systems.

Power Subsystem

The power subsystem will take 120 Volts AC as an input and output 5 Volts DC. The input power will be made usable by passing it through a 5V 2A AC/DC power supply. It will then power a switch whose state is determined via user input before finally going through a 2 A fuse. The resulting output power will then be distributed to the four other sub-systems. A 0.1 uF capacitor will also be integrated into the design to protect the circuit from high frequency noise.

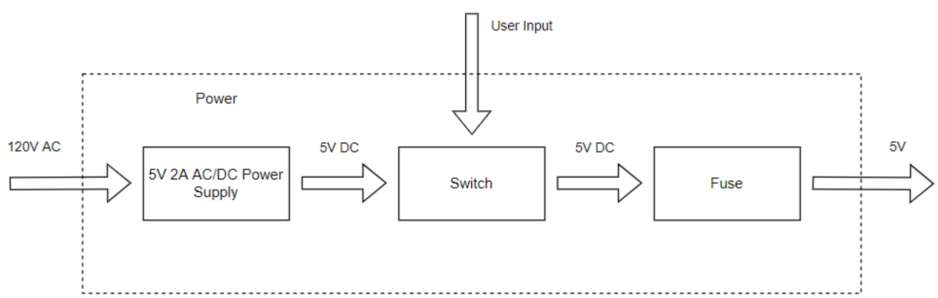

Figure 6: Power Subsystem

Audio Sub-System

The audio system will take 5 Volts DC and receive an analog signal from the processor via a 3.5mm auxiliary cable, then amplify it with an LM386 IC and finally play it through the speaker.

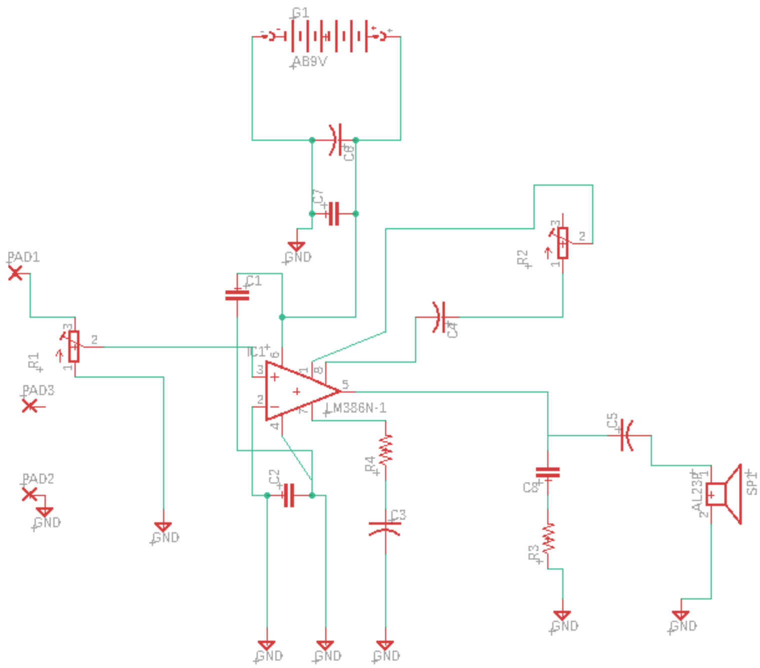

Figure : Audio sub-system circuit layout in Eagle

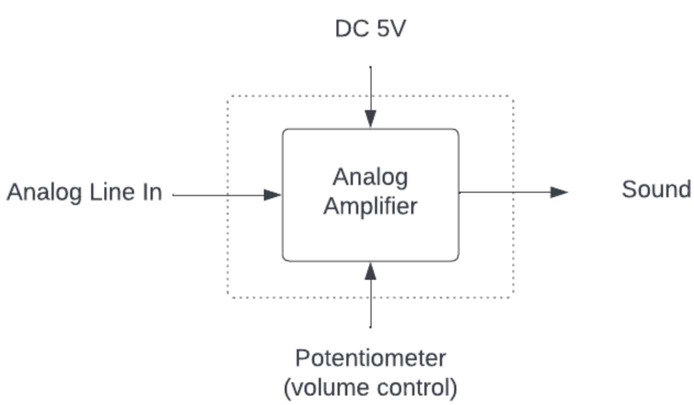

Figure 8: Audio sub-system block diagram

Mechanical Sub-System

The mechanical subsystem will take 5 Volts DC and receive digital signal lines through the Raspberry Pi to control the servo motor drivers. The motor drivers will then interpret the signals and power the motors appropriately to produce the desired 180-degree arm motion.

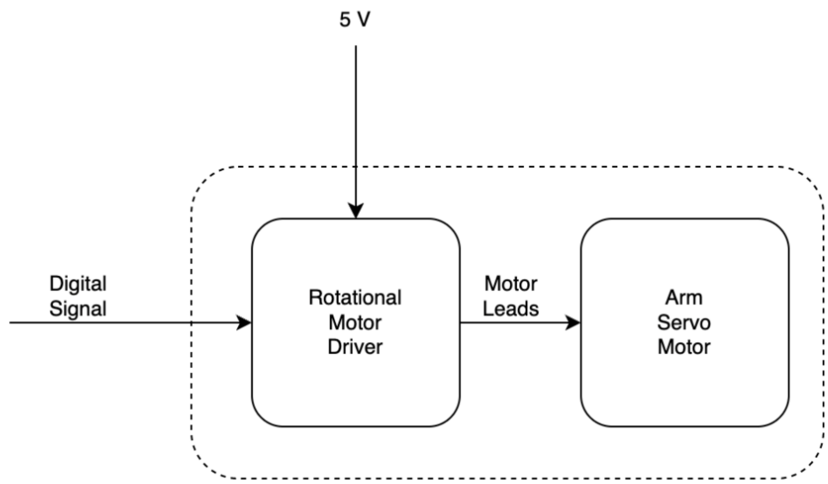

Figure 9: Mechanical Sub-System

Software/Processor Subsystem

The software subsystem will program the animatronic to perform tasks in either a test or director setting. In the test setting, the user will test out specific functions on the animatronic (I.e. arm mechanical test, audio test, LED test). In the director setting, the animatronic will be programmed to connect to the director and decode the director's CSV file to be ready to perform when commanded.

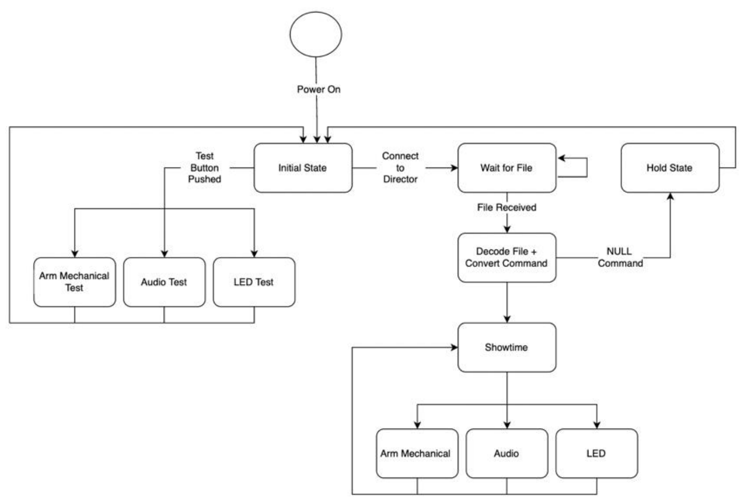

Figure 10: Software Sub-System Flow Chart

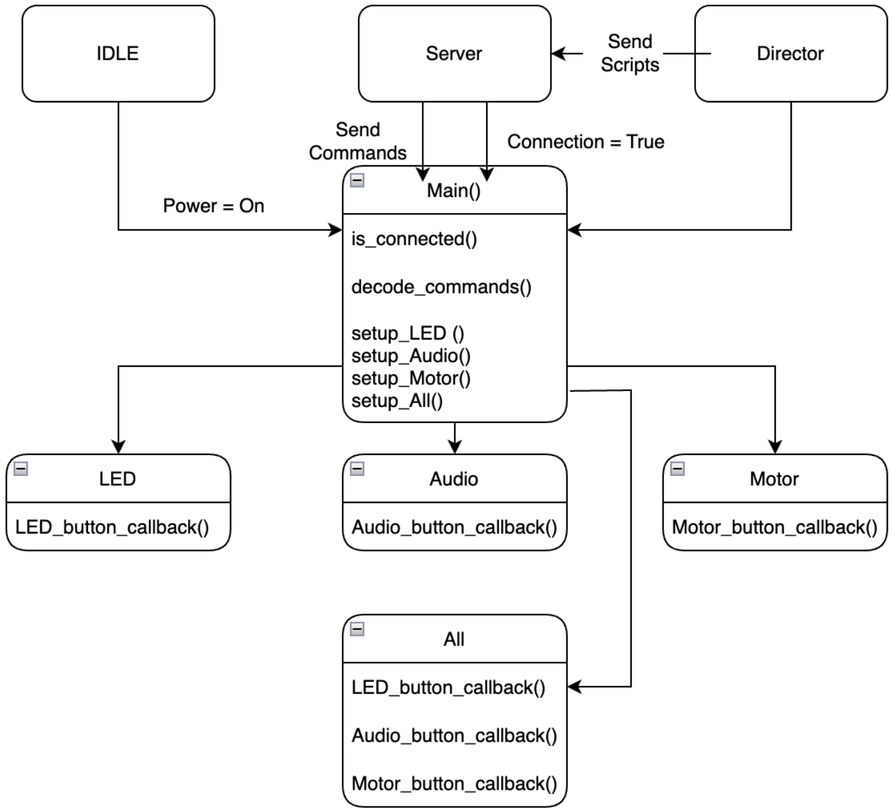

Figure 11: Software Architecture

Once the robot and director are connected, an infinite while loop will be entered. To provide users with a friendly user experience, one can press any of the four buttons on the robot thespian's box and the robot will come alive. Once a button is pressed the software will output its matching action. Figures 10, 11, and 112 show the code for when a button is pressed.

```
if GPIO.input(Button_Led)== GPIO.HIGH:
    print("Led Button was pushed! HIGH")
    GPIO.output(LED_Led, True)
    print("The LED has been switched ON.")
    sleep(3)
    print('3 seconds')
    GPIO.output(LED_Led, False)
    print("The LED has been switched OFF.")
else:
    print("led Button was not pushed! LOW")
```

Figure 10. Software for when LED Button is pressed

```python
if  GPIO.input(Button_Motor)== GPIO.HIGH:
    print("Motor Button was pushed! HIGH")
    print('Button 1 Pressed. Motor Demo!')
    p = GPIO.PWM(servoTailPIN, 50) # GPIO 12 for PWM with 50Hz
    p.start(0) # Initialization
    p.ChangeDutyCycle(2)
    time.sleep(1)
    p.ChangeDutyCycle(10)
    time.sleep(1)
    p.ChangeDutyCycle(2)
    time.sleep(1)
    p.ChangeDutyCycle(10)
    time.sleep(1)
    p.ChangeDutyCycle(2)
    time.sleep(1)
    p.stop(0)
    print('stop')
else:
    GPIO.output(LED_Motor, GPIO.LOW)
    print("Motor Button was not pushed! LOW")
```

Figure 10. Software for when Motor Button is pressed

```python
if GPIO.input(Button_Audio) == GPIO.HIGH:
    print('Button 2 Pressed. Audio Demo!')
    sound.play()
    print("done")
else:
    print("Audio Button was not pushed! LOW")
```

Figure 12. Software for when Audio Button is pressed

Light/Visual Subsystem

The light/visual subsystem will take 5 Volts DC and receive digital signal lines through the Raspberry Pi to control the LED wand. The Raspberry Pi will signal to the LEDs to light up once the audio plays so that motion and light are concurrent.

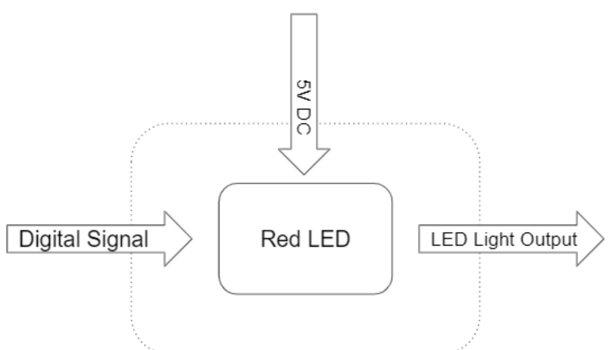

Figure 13: Light/Visual Sub-System

Constraints, Alternatives, and Trade-Offs

Mechanical:

- We had to create a custom 3D printed design to attach the servo motor to the doll base as the initial base was not strong enough to hold our servo.
- We redesigned our laser cut arm as our initial prototype was not large enough and the measurements for the servo attachment were incorrect.
- The servo motor was picked for the rotational motor because it is most accurate for our 180-degree range of motion.

Light/Visual:

- Initially we wanted to use higher voltage LED, like Neopixels, but the Neopixels required more power than supplied by our power supply, so we created a custom wand where a singular 3.3V LED could be seen. A pull-down resistor has been included to reduce the input voltage from the power supply.
- The initial design of the wand was created with the intention of hiding the wires for the LED in a hollow base, but once printed, the base was too small to fit the wires. Due to weight and proportion restrictions on the arm, the wand could not be printed larger. Instead, the wires are attached to the outside of the wand and are hidden with electrical tape.

Integration and Simulations

The most complex task of our project is integrating all the sub-systems together and ensuring that when they come together our robot performs as expected. To ensure the greatest chance of success, we need to simulate all our individual subsystems and make sure they pass the required tests.

Electronic Design

Power Subsystem

The power design features a 5V 2A AC/DC Power Supply connected with the user input-controlled switch and a fuse. Converting 120 V AC to the 5 V DC that will be used by all the subsystems is the general purpose of the design, in addition to determining the state via the switch and providing for safety considerations through the fuse and decoupling capacitor. The power fuse and decoupling capacitor are rated at 2 A and 0.1 uF respectively, protecting the circuit from both excessive current and high frequency noise. Note also the inclusion of a fuse holder to ensure easy replacement of the fuses. After individually testing out all of the physical components to confirm proper operation, work began on the overall subsystem simulation. Towards this end a schematic for the power subsystem was designed and implemented in Eagle. The results of the simulation proved the soundness of the design, converting 120 V AC to 5 V

DC with 2 A Current without issue. The next step in the process will involve integrating the power subsystem into the general system as a whole, and testing to make sure only 2 A of current are required for the full performance of each subsystem that depends on the input of the power supply. Pictured below is the power subsystem Eagle schematic used for simulation.

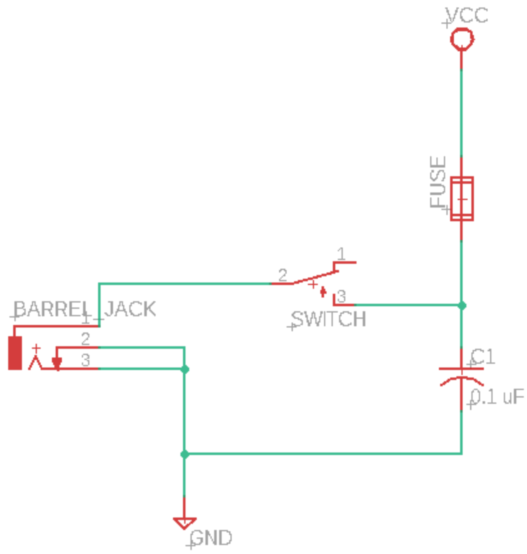

Figure 14. Power subsystem Eagle schematic

Audio Subsystem

The audio circuit features a LM386 audio amplifier IC. This chip requires 5-12V for power and functions similar to a standard op amp. Through simulating the device on a breadboard, we found that 5V created slightly quieter audio. After testing with a 9V battery, the audio was an appropriate level. Because of this, we will attempt to implement a boost converter to provide 9-12V to the PCB to increase customer satisfaction.

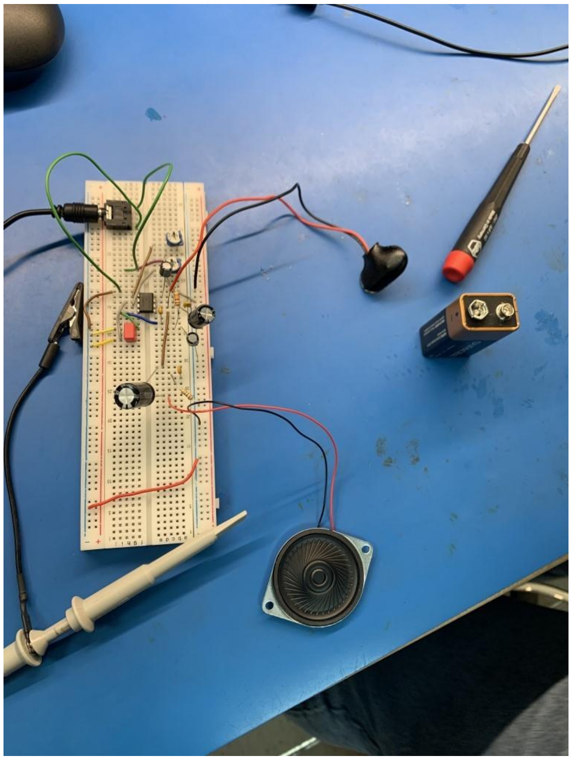

Figure 15: Audio subsystem simulation on a breadboard

Light/Visual Subsystem

The light and visual design incorporates an LED light in a 3D printed wand, wooden rectangular box as a base to hold all components, and barbie doll head with custom altered doll dress and torso made of clay and 3D print for structural integrity. The 3D printed wand is attached to the mechanical arm and is designed to be hollow to hide the wires attached to the LED. The box was created using boxes.py and is made from wood with an acrylic window to display the components inside. The box displays the buttons for user input and lasercut information and buzz logo. The dress will be altered from an existing doll dress to replicate the Queen of Hearts character dress using red and heart embellished fabric.

Figure 16: 3D Print Rendering for Heart Wand with Hidden Wires

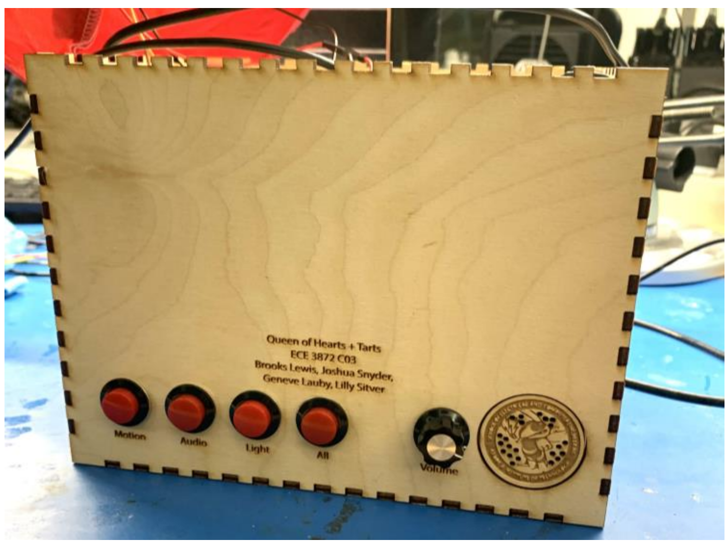

Figure 17. Laser Cut Wooden Box Design

Mechanical Design

Mechanical Subsystem

The mechanical design uses a servo motor to move the arm of our Queen of Hearts figure. The servo needed to be attached to the torso of the barbie doll base we are using for our Queen of Hearts figure, but when attaching the servo, the torso was too small. The laser cut arm was initially too small, so another arm was created to allow a size large enough to hold the wand. We decided to 3D print a new torso that holds our servo, allowing for seamless arm motion, as well as add a clay neck to give the animatronic a more anthropomorphic look. The servo will use 5V and be controlled by a PWM signal as displayed in the simulation below with the exception that in our design we will be using a Raspberry Pi Zero W instead of an Arduino.

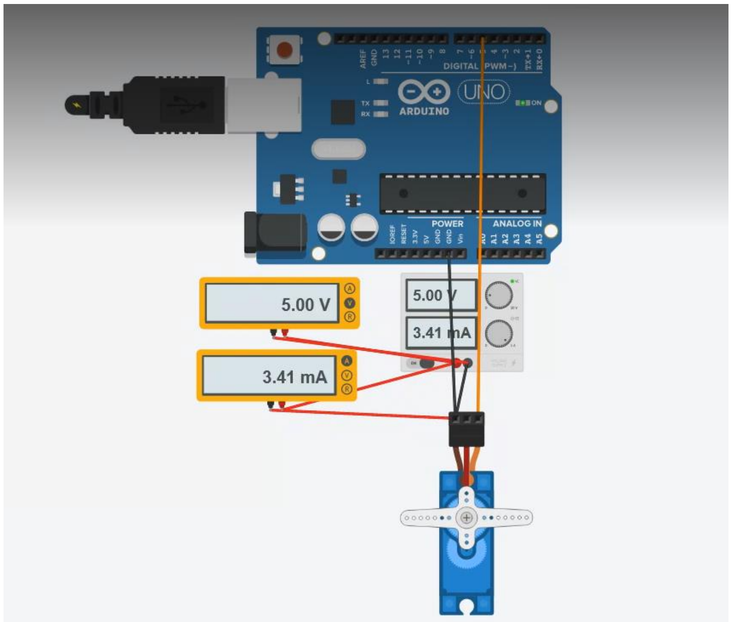

Figure 19: Mechanical subsystem simulation on tinkercad

Figure 20: Mechanical subsystem simulation on a breadboard

Software Design

Software/ Processor Subsystem

The software/processor design uses a Raspberry Pi Zero W processor to connect to the director and send digital signals to the light/visual, audio, and mechanical subsystems. These digital signals are PWM signals using any of the pi's GPIO pins. The Rasberry Pi has been connected to Wi-Fi and GTother. Establishing the connection between the raspberry Pi and director is still in progress. Our GitHub repository can be found at https://github.gatech.edu/lsitver3/ECE-3872-repo.

```
PING lilsit.local (172.20.10.11): 56 data bytes
64 bytes from 172.20.10.11: icmp_seq=0 ttl=64 time=25.310 ms
64 bytes from 172.20.10.11: icmp_seq=1 ttl=64 time=25.585 ms
64 bytes from 172.20.10.11: icmp_seq=2 ttl=64 time=17.321 ms
64 bytes from 172.20.10.11: icmp_seq=3 ttl=64 time=22.900 ms
64 bytes from 172.20.10.11: icmp_seq=4 ttl=64 time=45.494 ms
64 bytes from 172.20.10.11: icmp_seq=5 ttl=64 time=15.614 ms
64 bytes from 172.20.10.11: icmp_seq=6 ttl=64 time=14.271 ms
64 bytes from 172.20.10.11: icmp_seq=7 ttl=64 time=37.407 ms
64 bytes from 172.20.10.11: icmp_seq=8 ttl=64 time=14.196 ms
64 bytes from 172.20.10.11: icmp_seq=9 ttl=64 time=17.592 ms
```

Figure 13. Connection to Raspberry Pi

```
AttributeError: module 'keyboard' has no attribute 'wait'
KeyPress Proc finished
################################################################
Director Listening on ('127.0.0.1', 65432) for registration
################################################################


Finished Registration Process. Listing all registered robots
{}
Generating Robot Order List...
[]


################################################################


Initiating connection with registered robot order
Finished
```

Figure 21. Director running on local machine

Schedule

The schedule for completion of the project is shown in Table 2 where the task lead is indicated on each task.

Project: C03 Junior Design Schedule

	Week Number												
Task	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Brainstorm	Geneve	Geneve											
Design Visual Summary	Brooks	Brooks											
Design Top Level Block Diagram	Joshua	Joshua											
Design Software Block Diagram	Lilly	Lilly											
Proposal		All											
PDR					All								
CDR								All					
System Integration											All		
System Test											All		
Final Inspection and Demonstr												All	
Power Subsystem													
Design				Joshua	Joshua	Joshua							
Design Power Schematic				Joshua	Joshua	Joshua							
Obtain components					Joshua	Joshua							
Test component operation						Joshua							
Build							Joshua	Joshua	Joshua				
Integrate with subsystems							Joshua	Joshua					
Solder components									Joshua				
Test										Joshua	Joshua		
Test power output/safeguards										Joshua	Joshua		
Verify operation of other subsystems										Joshua	Joshua		
Audio Subsystem													
Design				Brooks	Brooks	Brooks							
Multisim schematic				Brooks	Brooks								
Breadboard circuit test					Brooks	Brooks							
Create Eagle schematic						Brooks							
Obtain components						Brooks							
Build							Brooks	Brooks	Brooks				
Obtain finished PCB							Brooks	Brooks					
Solder components onto PCB								Brooks	Brooks				
Verify PCB fits within enclosure									Brooks				
Test										Brooks	Brooks		
Mount PCB in enclosure										Brooks			
Play audio files through PCB											Brooks		
Software/Processor Subsystem													
Design				Lilly	Lilly	Lilly							
Software Architecture Diagram				Lilly	Lilly	Lilly							
Obtain Rasberry Pi					Lilly								
Build							Lilly	Lilly	Lilly				
Setup Headless Rasberry Pi Connection							Lilly	Lilly					
Connect to Director								Lilly	Lilly				
Interconnect Director, Robot, & Computer								Lilly	Lilly				
Test										Lilly	Lilly		
Simulate Software Connections										Lilly	Lilly		
Mechanical/Movement Subsystem													
Design				Geneve	Geneve	Geneve							
Render Arm to be Lasercut					Geneve	Geneve							
Design Connections - Servo to Processor				Geneve	Geneve								
Obtain Components						Geneve							
Build							Geneve	Geneve	Geneve				
Breadboard Circuit for Servo Function							Geneve	Geneve	Geneve				
Lasercut Arm and Attach Servo								Geneve					
Test									Geneve	Geneve	Geneve		
Visualize Arm Movement from Servo									Geneve	Geneve	Geneve		
Test Servo Function from Processor									Geneve	Geneve	Geneve		
Light + Visual Subsystem													
Design					Brooks	Brooks							
Render Custom Wand for 3D Print					Brooks	Brooks							
Design Base / Box to be Laser Cut					Brooks	Brooks							
Design Connections - LED to Processor					Brooks	Brooks							
Obtain Components						Brooks							
Build							Brooks	Brooks	Brooks				
Breadboard Circuit for LED Function							Brooks	Brooks	Brooks				
Lasercut Box and 3D Print Wand							Brooks	Brooks	Brooks				
Test									Brooks	Brooks	Brooks		
Visualize Light from LED									Brooks	Brooks	Brooks		
Test LED Function from Processor									Brooks	Brooks	Brooks		

Table 2: Schedule to complete Queen of Hearts (September 12 – November 6)

Integration

For the Queen of Hearts animatronic to work as desired, the subsystems must be connected cohesively in addition to working individually. To do this, we will subject the animatronic to a series of tests, both for the individual subsystem and for the interconnection between them, especially between processor and audio, light, and mechanical.

We first simulated our mechanical, light, and audio subsystem on a breadboard. Once this was simulated, we coupled these subsystems with the processor and simulated each action through the processor. We then integrated our processor with the pushbuttons, speaker, and motor. The hardware and software was integrated together for our audio PCB.

Software High-Risk Parts:

The Raspberry Pi zero is the most high-risk part of our design because it is crucial to every sub-system in our project and controls the entire animatronic. In order to ensure that the processor works, we tested the processor and the software with each subsystem as well as the buttons. We first tested each subsystem on a breadboard before connecting the subsystem to the processor to ensure the component worked and the software was not faulty.

Hardware/Mechanical High-Risk Parts:

The servo motor is the most high-risk part of our hardware design as the servo requires a large amount of current and must function properly for the animatronic to work as intended. If the servo motor does not work, the arm will not move.

We had multiple PCBs made to test and solder components to. To reduce risk, we created the speaker on a breadboard and tested all components before soldering to the PCB. This allowed for confidence in our PCB design once we put the audio PCB together. Our PCB once soldered worked as intended.

To accomplish the light requirement, we created a custom 3D printed wand with the capability for an LED to be strung through the hollow base. The wand base was too narrow for the wires to fit through, so to mitigate this, we strung the LED wires around the wand instead.

Repository Management:

We have a GitHub that includes our electronic system files. GitHub also includes our software code. We also share files in our central Microsoft Teams SharePoint folder.

Conclusion

Over the last 13 weeks, our team was able to successfully build our animatronic, The Queen of Hearts + Tarts. The project was broken down into five subsystems, enabling us to delegate tasks and focus on separate components before integrating the subsystems into our final project. There were several challenges our team faced and overcame. Our first challenge was connecting the Raspberry Pi to the director as well as setting up the Raspberry Pi to be compatible with the GT network. This was resolved by taking out a part of the initial sample code that was incompatible with MAC. For the mechanical portion, there was another challenge as the servo was rotating backwards at 180 degrees. A change to the code fixed this as well. The light and visual subsystem encountered a problem with the wand design as the initial idea was to have the wires to the LED be fed through the wand base, but the hollow wand diameter was too small. By testing each subsystem separately, we could resolve problems early, making the ultimate interconnection between subsystems smoother.

If given this project again, there are a couple of things our team would do differently. The first thing we would have changed would be following and creating a schedule that accounts for problems in the design and the software. When deciding on the doll base, we realized we needed a base that would be able to hold the servo, which took some time to create and 3D print. Similarly, the hollow wand was too small for two wires to fit through, so we had to change our plan as well. With some more forethought into these problems, such as accounting for the long 3D print times, we may have been able to mitigate these obstacles more strategically, but overall, this did not affect our final product and resulted in a successful animatronic that accomplished all required tasks.