The City College of New York
Grove School of Engineering

Project: Nala, A Smart Companion for Seniors

Department of Electrical Engineering
EE 59867: Senior Design II, Spring 2018

Mentor: Professor Xiao
Instructor: Professor Baurin
Project Manager: Randy Garcia
Project Members: Hongyu Shi, Parmanand Shiwmangal, Thomas Wilk, Mohammad Zilon

Date: May 16, 2018

Table of Contents

Abstract

A companion bot named “Nala,” capable of appointment reminders, memory games, and LED matrix visual response was built. The exterior of the bot’s body is shaped like a friendly cat, which is meant to reduce anxiety of the user, and encourage the user to be engaged mentally, even during extended periods of physical isolation. Using Raspberry Pi embedded hardware, and the Amazon Alexa API, the bot is able to interact through voice commands from the user. The robot’s body has been designed with Fusion 360 and is made to sit on a table near the user.

Introduction

We propose building a tabletop-sized, artificially-intelligent senior citizen companion robot. The device will prompt its user with reminders for events, such as taking medications. The device will be able to play memory/information recall games, allowing the user to practice memory skills or to have fun. Lastly, the external appearance of the robot will be of something that looks pleasant, cute, or fun. The robot will provide life alerts, reminder functions and functions for fun. The machine-learning nature of the robot will allow the user to create useful alerts, and reminders. The overall result of the robot will be a device that serves as a support companion to the user, which could provide life-saving and therapeutic value.

Need

Many senior citizens in the United States report feeling lonely or depressed. Whether living at home or in nursing home, large numbers of these older adults report feeling isolated, even if they have regular contact with family members. And beyond the adults that do have families to care for them, there are still large numbers of adults who do not have families. Many nursing homes and senior centers in the United States are short-staffed due to budget constraints, and are not able to adequately supervise all of their residents. Additionally, although senior centers provide recreational activities to seniors, there are still seniors that remain isolated, and some seniors report not speaking to anyone during a single day.

Overall, in cases where older adults cannot have the care they require, they are faced with unmet needs of social interaction, medical supervision, and healthy structured schedules. In a study done in Selwyn Heights retirement home in New Zealand a group of 40 seniors participated in this study to see if the Japanese robotic companion “Paro” would reduce feelings of loneliness and depression and improve their quality of life. Their feelings of loneliness were measured using the UCLA loneliness scale 3; depression was measured with Geriatric Depression Scale (GDS). At the end of 12 weeks, the feelings of the elderly residents were measured and compared to their initially recorded feelings, with the results showing that

loneliness and depression in the group who interacted with Paro was indeed reduced compared to the control group without Paro [1].

Additionally, in some cultures it is strongly encouraged to care for adults at home and to not send these adults to live in adult care facilities. Some families may also choose to hire a stay at home nurse, but this is only available to those who can afford it. Many families want to take care of their own family members, but in many households, the entire family has to work to make enough money to support itself. Therefore, their parents or grandparents are left unattended.

Our senior companion device will not replace nurses or families taking care of their elders. Instead it will provide support to the nurses, as well as families. This device could be used at home or in nursing homes. It will be a lifesaving and fun companion to seniors while helping and entertaining them.

Table 1.1: Selwyn Heights Paro Study Results [1]

Primary Psychosocial Outcomes

	Paro	Control	*F* Test for Group X Change Since Baseline			
	Adj Mean (SD)	Adj Mean (SD)	*F*	*df*	*P*	η_p^2
Quality of Life						
Baseline (T1)	33.94 (7.51)	33.42 (6.99)				
Follow-up (T2)	32.73 (8.24)	31.19 (6.26)				
Change score	−1.33 (5.77)	−1.88 (4.27)	0.22	1, 28	.64	0.01
Staff rated Quality of life						
Baseline (T1)	31.15 (6.70)	32.05 (8.83)				
Follow-up (T2)	26.71 (7.71)	23.94 (5.18)				
Change score (T2−T1)	−5.71 (7.65)	−7.06 (8.36)	1.18	1, 31	.29	0.04
Depression						
Baseline (T1)	4.88 (3.58)	3.33 (3.22)				
Follow-up (T2)	4.15 (2.34)	4.00 (2.62)				
Change score (T2−T1)	−.64 (3.89)	.40 (2.56)	0.00	1, 26	.97	0.00
Loneliness						
Baseline (T1)	36.44 (9.76)	31.71 (9.50)				
Follow-up (T2)	32.23 (9.92)	33.93 (8.52)				
Change score (T2−T1)	−5.38 (7.58)	2.29 (6.19)	5.14	1, 24	.03	0.18

Problem

The Smart Companion for Seniors will target the older generation, be it at home or at a senior center. The main problem addressed will be companionship for the elderly. The companion's size, pleasant appearance, motion degrees of freedom, and voice interaction capability will help to alleviate those feelings of the user being isolated, lonely or bored. As mentioned above, most adults are out in the working world while their elderly parents are often left alone or are at a senior center where they might not be given adequate attention. This is where our system can play an important role. The initial model will be designed for a desktop for close contact with its user. Our design will go further and address some common requirements of

the elderly. These needs include a the ability alert someone in case of emergency, play soothing music, create reminders and todo list, play memory games and other games for entertainment, and listen to stories or the news.

ELECTRICAL SYSTEM

Block Diagram of Electrical System

The diagram below outlines the general flow used in the development of Nala's electrical aspect. Each of the components play a key role. It should be noted that these components were bought pre-assembled then modified and wired accordingly.

- The raspberry pi 3 was used as the brain of the bot. It supports the necessary peripherals together with amazon's alexa.
- The microphone functions as the receiver to capture the voice of the user and pass it onto the RPI where it is processed by alexa.
- The speaker serves as the output for alexa's voice.
- The amplifier allows for the increase and decrease of audio level to the speaker.
- The ground loop noise isolator assists with removing unnecessary wall outlet frequencies.
- The headphone jack port can be used for private listening.
- The 16x32 rgb led panel portrays the two lit eyes of nala.
- The power button allows the RPI to be on/off.
- The 5V/3A supply powers the RPI
- The 12V/10A supply powers the amplifier as well as the matrix panel
- The buck converters reduces the supply voltage to required voltage for device.
- 3A fuse aids in protection for the 16x32 rgb led panel

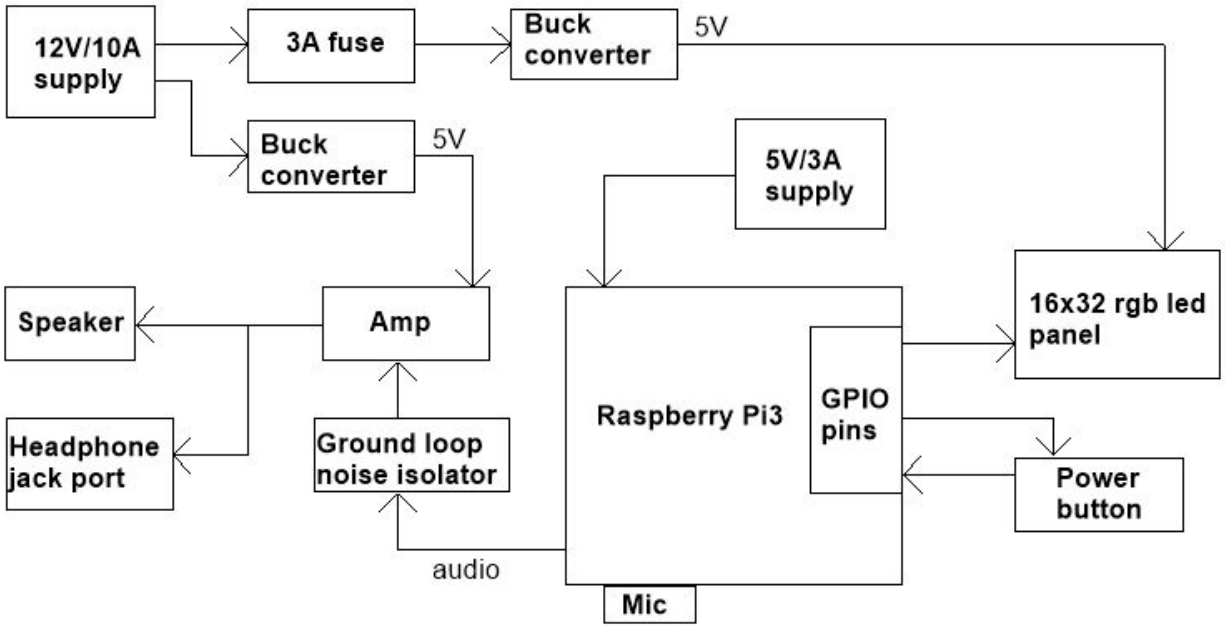


Figure 2.1: Block diagram of electrical system.

Raspberry Pi 3

The Raspberry Pi 3 (RPI3) is a pretty powerful computing device for being a computer that is the size of a credit card and weighs about 42 grams. This is why we choose this tiny computer to be the brain of our senior care companion. The processing power from the Raspberry Pi 3 Model B comes from its 1.2 GHz quad-core 64-bit ARM Cortex A53 processor. The pi has 1GB of LPDDR2-900 SDRAM, it also has a 400 MHz Dual Core VideoCore IV GPU that is capable of 1080p HD playback. In terms of audio it includes two forms of audio output the first being a HDMI port and the second a 3.5mm headphone jack, the same HDMI port can also function as a video output.

In terms of communication it has Ethernet support, 802.11n WiFi wireless Networking, and Bluetooth 4.1 wireless technology. The raspberry pi contains many connections which include four USB 2.0 ports, DSI Display port, CSI Camera port, micro SD card slot, and 40 GPIO pins. To power this tiny computer an input voltage of 5V DC, with a current of 3.0 Amps to 2.5 Amps is required. The raspberry pi also compatible with a plethora of operating systems Libreelec, Open Elec, OSMC, and Pinet being a few. For our usage we will be using the officially supported operating system NOOBs. For more details refer to Appendix B.

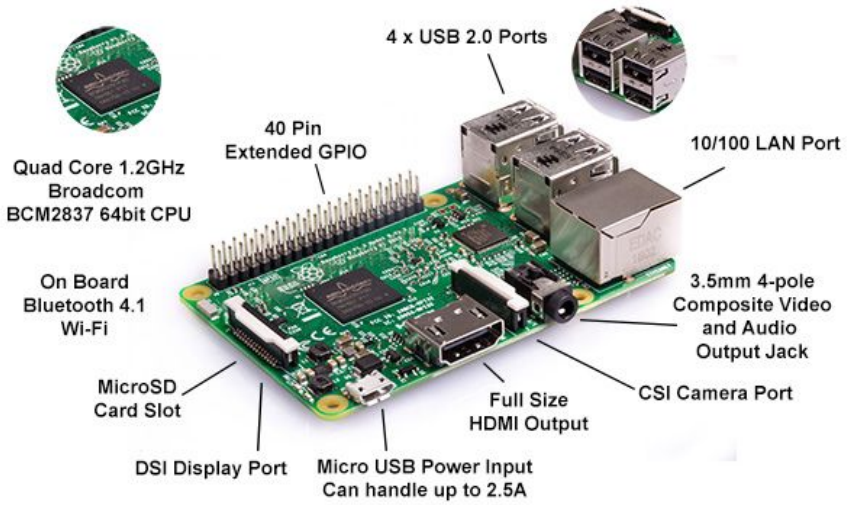


Figure 2.2: Ports/connections of the Raspberry Pi 3

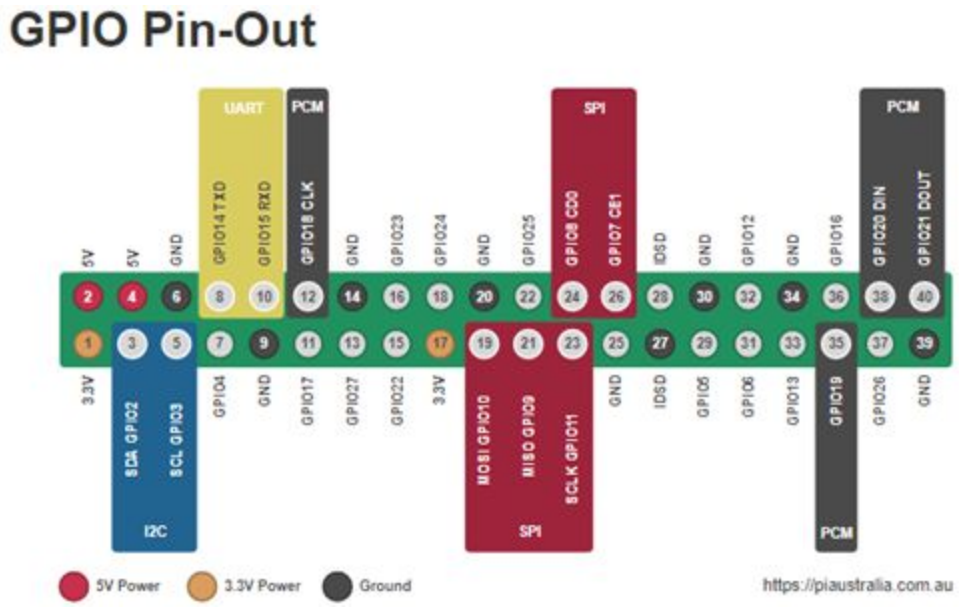


Figure 2.3: GPIO pin-out layout of the Raspberry Pi 3

To get the raspberry pi up and running a couple of items are needed the first being the raspberry pi itself, a monitor with an HDMI ports, a keyboard, mouse, micro SD card, a power supply, and a laptop. The micro SD card must be larger than 4GB to contain the software files and so an 8GB micro SD was used. Once the micro SD is formatted, the NOOBS folder can be download from the raspberrypi.org onto the laptop.

After the NOOBS folder is downloaded, all the files in the NOOBS directory are copied and then pasted into the micro SD card which will take about 12-15 minutes. The raspberry pi is then connected to the keyboard, mouse, monitor and the micro SD card is inserted. Once all of that is done the power supply can be connected to boot the raspberry pi. A window with many operating software will open, select raspbian and click install this will take some time.

Once the installation is complete the graphical user interface will installed, we will have access to the raspberry pi's desktop environment.

The first thing to do is to open the terminal and run the commands which will ensure that the raspberry is updated to its latest version and that Java is installed which is essential for running the Alexa Sample App:

1. sudo apt-get update
2. sudo apt-get install oracle-java8-jdk

So that raspberry pi could be controlled remotely and without a monitor, keyboard or mouse we turned VNC which is a graphical desktop sharing systems that allows a user to control a computer remotely.

It was done so by navigating to :

-  ------> Preferences ------> Raspberry Pi Configuration
- This will open a new window, interfaces was selected
- And VNC was then enabled

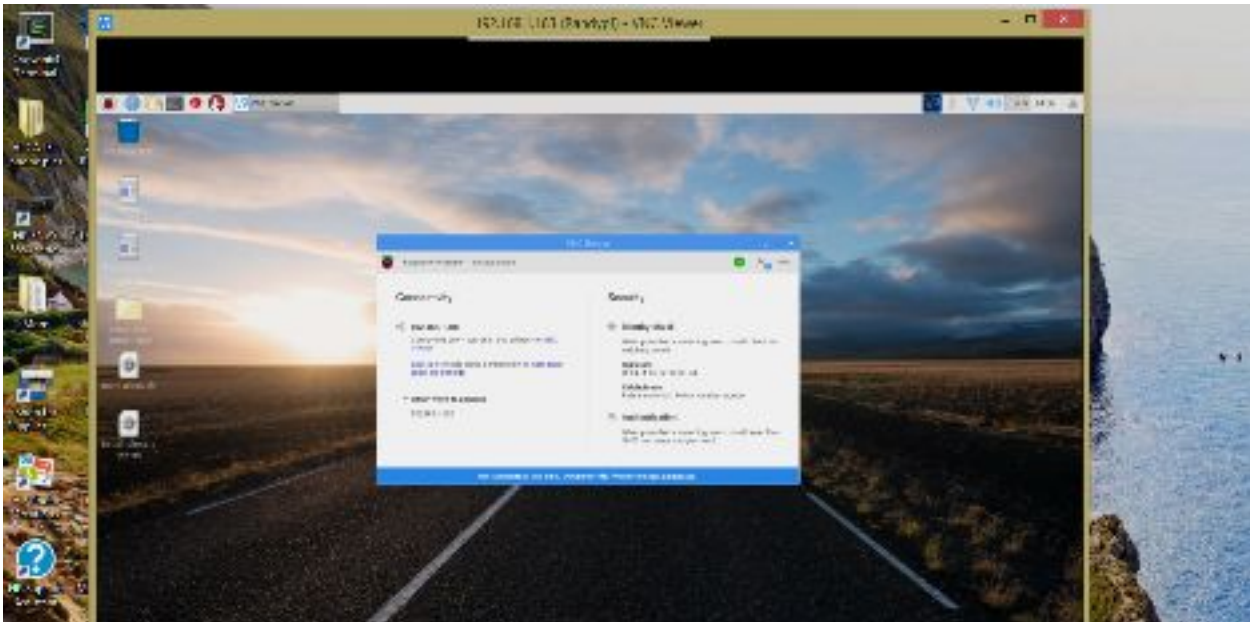


Figure 2.4: VNC control of the Raspberry Pi 3

Microphone + Speaker + Amplifier + Audio Jack port

The combination of these four components works in conjunction with the Alexa voice service that is installed on the RPI. The microphone a mini usb type (see appendix B for device details). It captures the voice of the user and relays it to Alexa for interpretation. The mic is connected to one of the onboard USB ports of the RPI. The response provided is heard through the speaker. A 4 Ohms 5W speaker was used (see appendix B for device details).

In order to have a louder and controlled sound, an amplifier was included. We used the TDA2822M amplifier which is rated at 1.8-12V 5W maximum (see appendix B for device details). It operates by taking the input and multiplying it by a amplification factor or gain to yield an increase output as shown mathematically by: Vout = Gain x Vin
The amplifier also houses a potentiometer which allows the sound to be adjusted as desired.

The audio jack port is there if the user wants to listen privately (see appendix B for device details). It is wired in such a way that when the headphone jack is inserted into the port, the connection from the amplifier to the speaker is interrupted and the signal is sent to the headphones.

After testing Nala we noticed a loud hum coming from the speaker. Analysis showed that this was a result of using the 16x32 LED matrix in conjunction with audio. Turns out that both of these need to utilize the pulse width modulation (pwm) capabilities of the raspberry pi. As a result, using them at the same time gives poor audio quality. To improve the sound, a ground loop isolator was connected between the 3.5mm output of the RPI and input of the amp (see appendix B for device details) . The isolator helps to remove unwanted 50/60Hz frequencies coming from the supply which in turn received it from the wall outlet. This isolator works in conjunction with a decoupling capacitor across the 12V/10A power supply line and grounding RPI GND floating pins to drastically removed the noise from the speaker. By doing this to the ground pins of the RPI, the voltage reference level of the device is better stabilized.

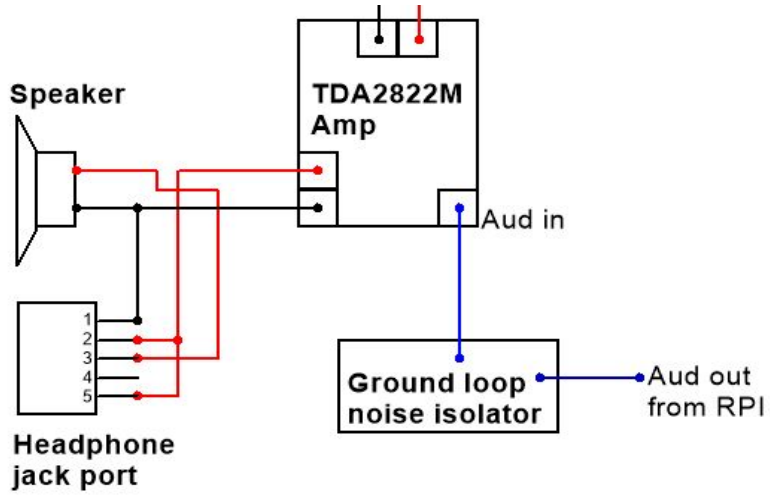


Figure 2.5: Wiring Schematic of Speaker, Headphone jack and Amp

Below is an outline of the steps followed to set up these components:

- The mic was plugged into the USB port and the driver automatically installed.

- To improve the capabilities of the mic the following was done:

  - Go to  ------> Preferences ------> Audio Device Settings
  - From the drop down menu ‘USB PnP Sound Device (Alsa Mixer)’ was selected
  - Select controls -----> check box for ‘Microphone’ and ‘Auto Gain Control’
  - Switches -----> check box for ‘Auto Gain Control’

- The audio jack port and speaker were wired together as shown in the schematic after which they were hooked to the amplifier.
- Finally the isolator was connected between the audio output of the RPI and input to the amplifier.

Power Button

Many electrical devices carry a power button that serves not only to turn the device on/off but also to protect it, conserve energy and provide convenience. However the raspberry pi class of microcontrollers does not come with one and this might have been due to a cost factor or was just not included in the system architecture. With this in mind the current means of shutting down is by following the Graphical User Interface (GUI).

 --------> Shutdown -------> Confirm

Or it can also be down through the command line using the following: ‘sudo shutdown -h now’

But these means are not convenient especially if there is no display present, RPI cannot be accessed remotely or is installed in a consumer product as in our case. There are a few workarounds to include your own button. In this project two methods were taken into consideration. It is to be noted that the choice of switch used is that of a momentary kind where pressing it bridges the connection for a short time (see appendix B for device details).

The first of these workarounds was a combination of hardware and software and included a single switch which was placed across GPIO 3 (pin 5) and ground. GPIO 3 is unique since according to the Raspberry Pi website it provides a “useful way to talk to many different types of external peripheral” [https://pinout.xyz/pinout/i2c]. This pin is normally pulled up to a voltage of 3.3V and when pulled down to ground using the switch the status changes. If the RPI is in a halt state pressing the button would wake the pi and allow it to boot. On the other hand if the RPI is already on, bridging these two pins would produce no result and that is where the software aspect comes in. A shutdown script was needed.

The second method included two switches, one for on and another for off. To turn on the Pi we made use of the P6 pads or 'RUN' pads on the circuit board. The 'RUN' pads essentially allows the device to be reset without having to unplug the power supply. With the PI in a halt state, momentarily shorting these pins allowed for booting. If however the PI is already on, shorting would simply cause a reboot instead of shutting down the device. This is why a second button was needed. A shutdown script had to be written for a GPIO pin to ground connection similarly to the that used in method one above.

Out of the two methods, method one was chosen because it uses only a single switch and seemed more convenient to our application. Below is an outline of the steps followed to get the power button in place.

- The circuit was built. Two wires were attached to the switch and then connected to GPIO 3 and ground. In this case the positive/negative terminals of the switch had no effect. Either wire could be used on GPIO 3. Having the RPI in a halt state the button was then tested.

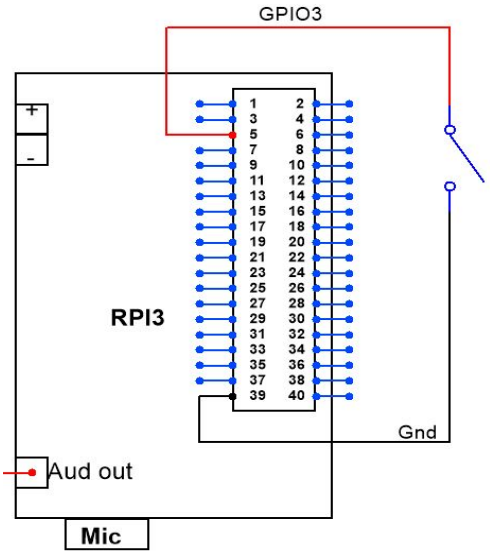


Figure 2.6: Wiring Schematic of Power button

- The shutdown script was then created with the help of the following sources: Tyler, "How to add a power button to your raspberry pi" & AndrewH7, "Simple Raspberry Pi Shutdown Button"
  - From the command line of the RPI the Nano text editor was open: 'sudo nano'
  - The script was edited, compiled and saved. The full script is shown below.

```
#!/bin/python
import RPi.GPIO as GPIO
import subprocess
GPIO.setmode(GPIO.BCM)
GPIO.setup(3,GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.wait_for_edge(3,GPIO.FALLING)
subprocess.call(['shutdown','-h','now'],shell=False)
GPIO.cleanup()
```

Figure 2.7: Shutdown script

- Next it was set to be initialized upon boot. The local file was open: 'sudo nano /etc/rc.local'
- 'python /home/pi/power.py' was then added before 'exit 0' in the local file
- The file was made executable: 'sudo chmod +x /home/pi/power.py'

- Once the script was set, the RPI was rebooted and the switch was tested to verify that both the off and on function works.

16x32 RGB LED Matrix

We have used a 16x32 LED (Light-emitting diode) Matrix panel for Nala to visually interact with the user. Through the Pi's GPIOs (general-purpose input/output pins) we have full control of the 16x32 LEDs. The Panel requires a 5V constant DC Voltage with maximum 3A current delivery. We are using a buck converter to regulate the voltage. In addition we are using a 3A Fuse to protect the LED's from blowing.

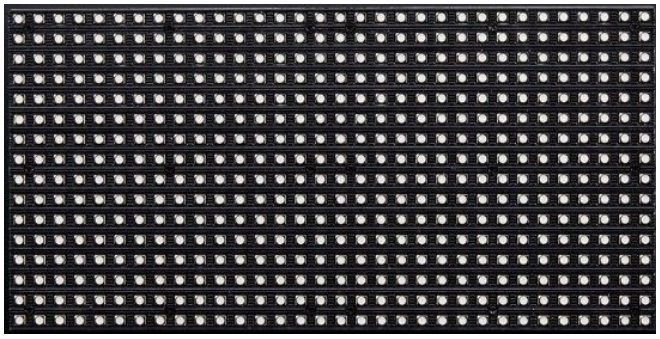


Figure 2.8: LED Matrix Panel.

The Technical Details of the LED Matrix is in Appendix B.

Equipments for the LED Panel:

- Jumper wires

- Ribbon cable
- 5V Power supply with 2-3A current regulator.

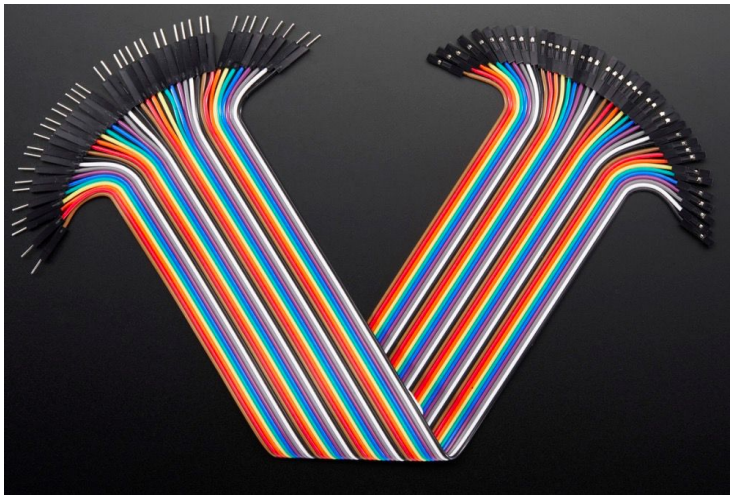


Figure 2.9: Jumper wires

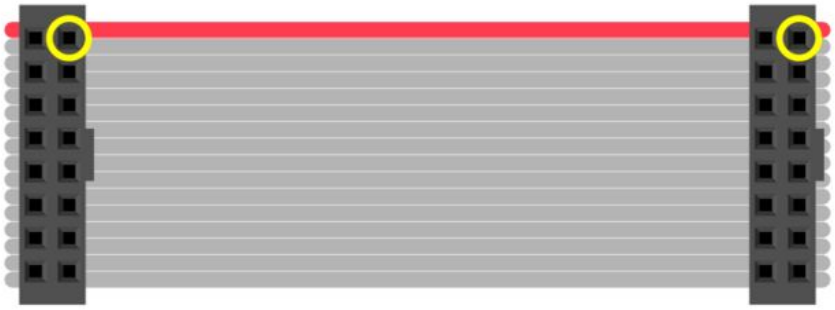


Figure 2.10: Ribbon cable.

Connecting the LED panel to the Pi's GPIOs:

First, we connect the ribbon cable to the LED Panel Input Pins Block as shown in the figure below.

Figure 2.11: Ribbon cable connected the LED Panel Input Pins Block

The wiring scheme used to connect the GPIO pins of the pI to the Input pins of the LED Panel is as follows:

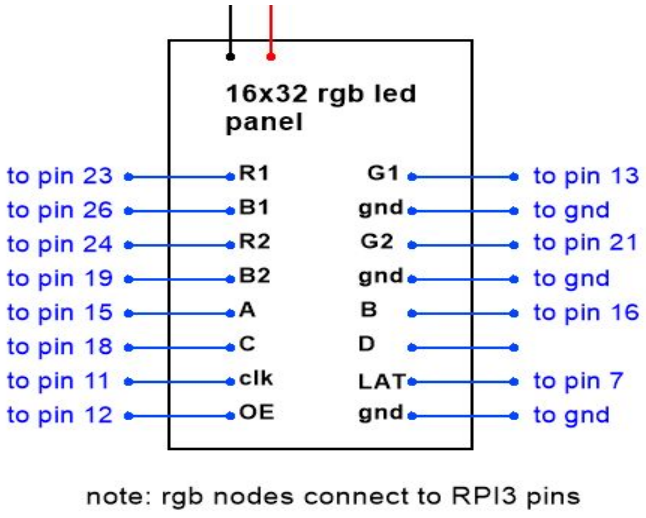


Figure 2.12: Wiring Schematic of 16x32 RGB LED panel

The connections in a list format:

- GND on display to GND on the Pi
- R1 on display to GPIO 23 on the Pi
- G1 on display to GPIO 13 on the Pi
- B1 on display to GPIO 26 on the Pi
- R2 on display to GPIO 24 on the Pi
- G2 on display to GPIO 21 on the Pi
- B2 on display to GPIO 19 on the Pi
- A on display to GPIO 15 on the Pi
- B on display to GPIO 16 on the Pi

- C on display to GPIO 18 on the Pi
- OE on display to GPIO 12 on the Pi
- CLK on display to GPIO 11 on the Pi
- LAT on display to GPIO 7 on the Pi

Powering the LED Panel:

Now that we have connect the LED Panel to the Raspberry Pi, we have to power the LED Matrix. We will use a buck converter power source to power the display. The buck converter will be discussed in details in the power supply section of this report.

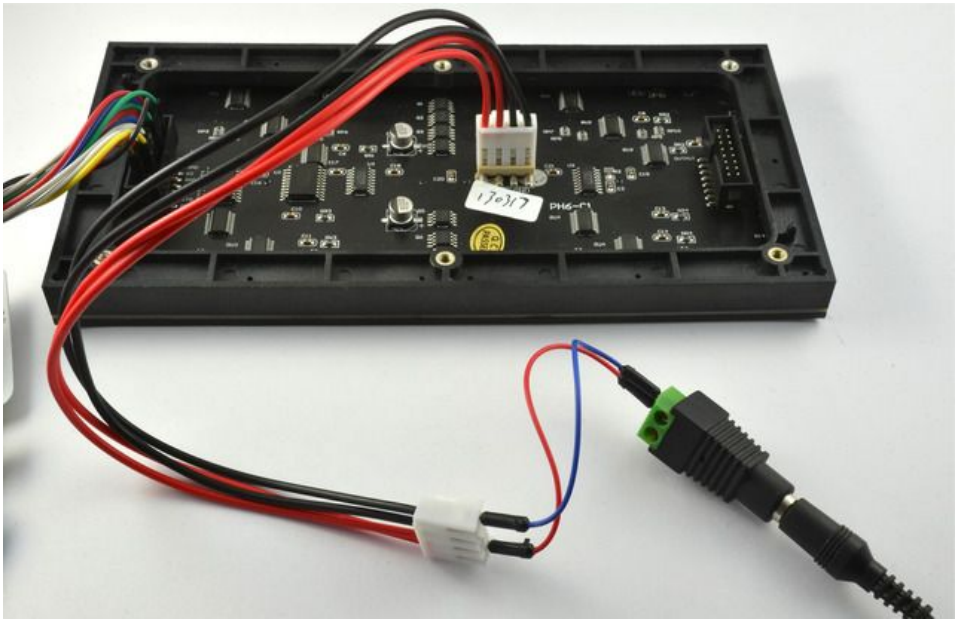


Figure 2.13: LED Panel powered by the buck converter.

Controlling the LED Panel via Python or C/C++ Code:

The Display can be controlled by one simple python code. The Python Code can be found in Appendix C.

C/C++ Code:

To use the C code, this Hzeller's Git [https://github.com/hzeller/rpi-rgb-led-matrix] has to be downloaded and be made on the Pi. The instruction are on the readme.md file.

Running Commands:

After the making the "Demo" file, we can run various demos. For example to run a rotating square script we have to run the following command via the Pi's Terminal app.
"sudo ./demo -D0 --led-no-hardware-pulse --led-cols=32 --led-rows=16 --led-chain=1 --led-brightness=50"

There is also an API to create a program to display an image or video. The link to the API is https://github.com/hzeller/rpi-rgb-led-matrix/tree/master/utils.

Display's Capabilities

Basic Functionalities:

- Scrolling and still Text.
- Image
- Video

The LED Matrix can display:

- Time
- Reminders/Notifications
- Monitored health status.
- Alexa's Feedback text.

Figure 2.14 : LED Matrix displaying Rotating Squares

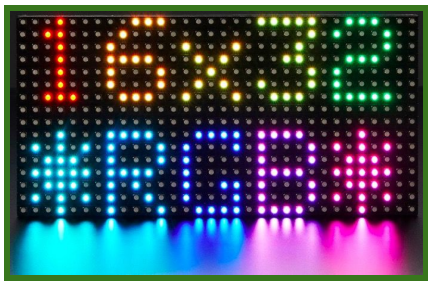


Figure 2.15: LED Matrix displaying a still text

Figure 2.16: LED Matrix displaying a scrolling message

Overview of how it all works:

First, the Java App starts and connects to the Alexa Voice Service, where the machine learning algorithms occur. Then a consecutive code runs that sends electrical signals to the LED Matrix via the Pi's GPIO pins. The Pi controls the LEDs to display images or texts on the LED Matrix. Simultaneously, other processes run such as the speaker letting the user know that the bot is ready. All the software and hardware modules are automated in the bot; therefore, the user does not need technical skills to interact with Nala.

Power Supply

Power supplies are necessary to convert the wall outlet ratings to that which is suitable for the load. They also convert alternating current (from the wall outlet) to direct current which the device operate on. Primarily there are two types, regulated and unregulated supplies. Regulated supplies maintains its rated voltage regardless of the current drawn or if there is a change in the input voltage. Keep in mind that there is a maximum value of current that can be tapped into and if gone over, the supply will behave improper. Unregulated supply are made to hold a steady power level and as such if more current is drawn the voltage will drop to keep the power level at balance. This follows the general equation: Power = Voltage x Current. For our project regulated supplies were used.

The initial plan for Nala, was to use the one supply that powers the raspberry pi and connect the amplifier and rgb led panel to its GPIO pins which can supply some power. However, when this was tested it was found that the RPI was being robbed of power and did not boot at all or if it did behaved abnormal. In another case, we got a higher rated supply and connect the RPI, amplifier and led panel in parallel to it. This worked but a constant loud hum was heard from the speaker. To remedy this we tried a high pass filter which blocks a range of

frequency below a cut-off frequency. The hum was drastically reduced from the speaker but since we are using a headphone, the hum is still heard when placed to the ear. This is not very user friendly.

The final decision was to use two supplies. The raspberry pi had its own while the amplifier and rgb led panel fed off another. The RPI was powered with a 5V/3A which is suitably rated and was connected directly to the port. The other peripherals fed off of a 12V/10A supply. Both the amp and led panel are rated at 5V, but supply of 12V was used because it was more readily available at the time and we needed at least 4A. To meet the specification of 5V, LM2596s buck converters were used to breakdown the voltage (see Appendix B for device details). To help with heat dissipation, small heat sinks were added onto the converter's ICs. A 3A fuse was also added to the positive line of the led panel's channel as a precaution since this device tend to draw high current. The 1000 micro-farad capacitor serves as a decoupling one to help reduce high frequency noise coming from the power supply. This aided in the hum reduction discussed in the Microphone + Speaker + Amplifier + Audio Jack port section above.

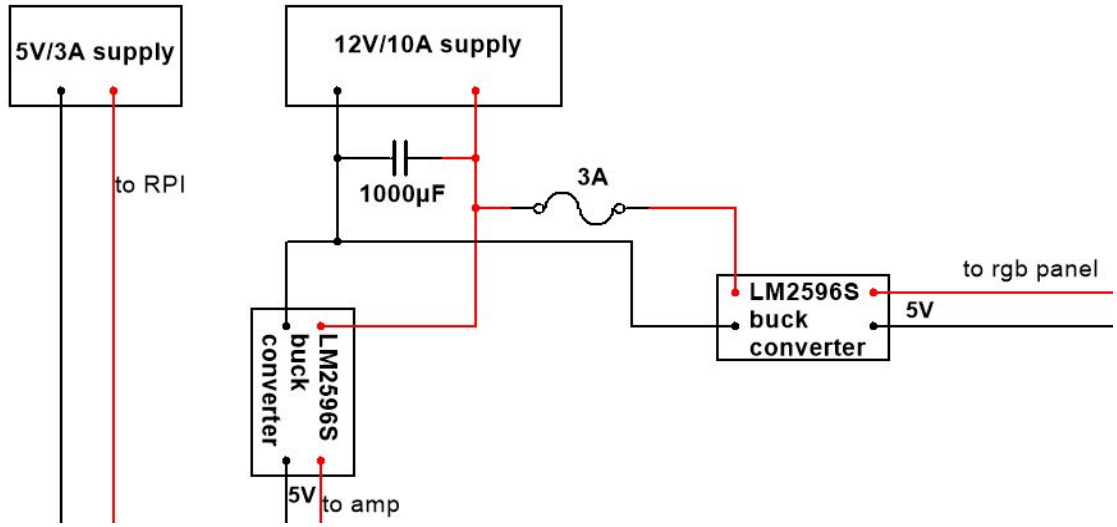


Figure 2.17: Wiring schematic of power supply

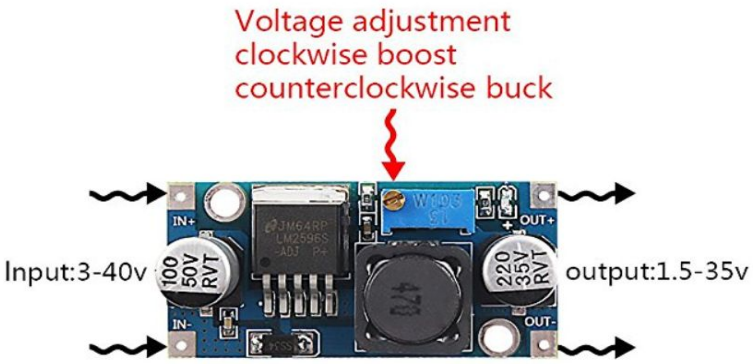


Figure 2.18: LM2596S DC-DC Buck Module

Complete Wiring Diagram of Electrical System

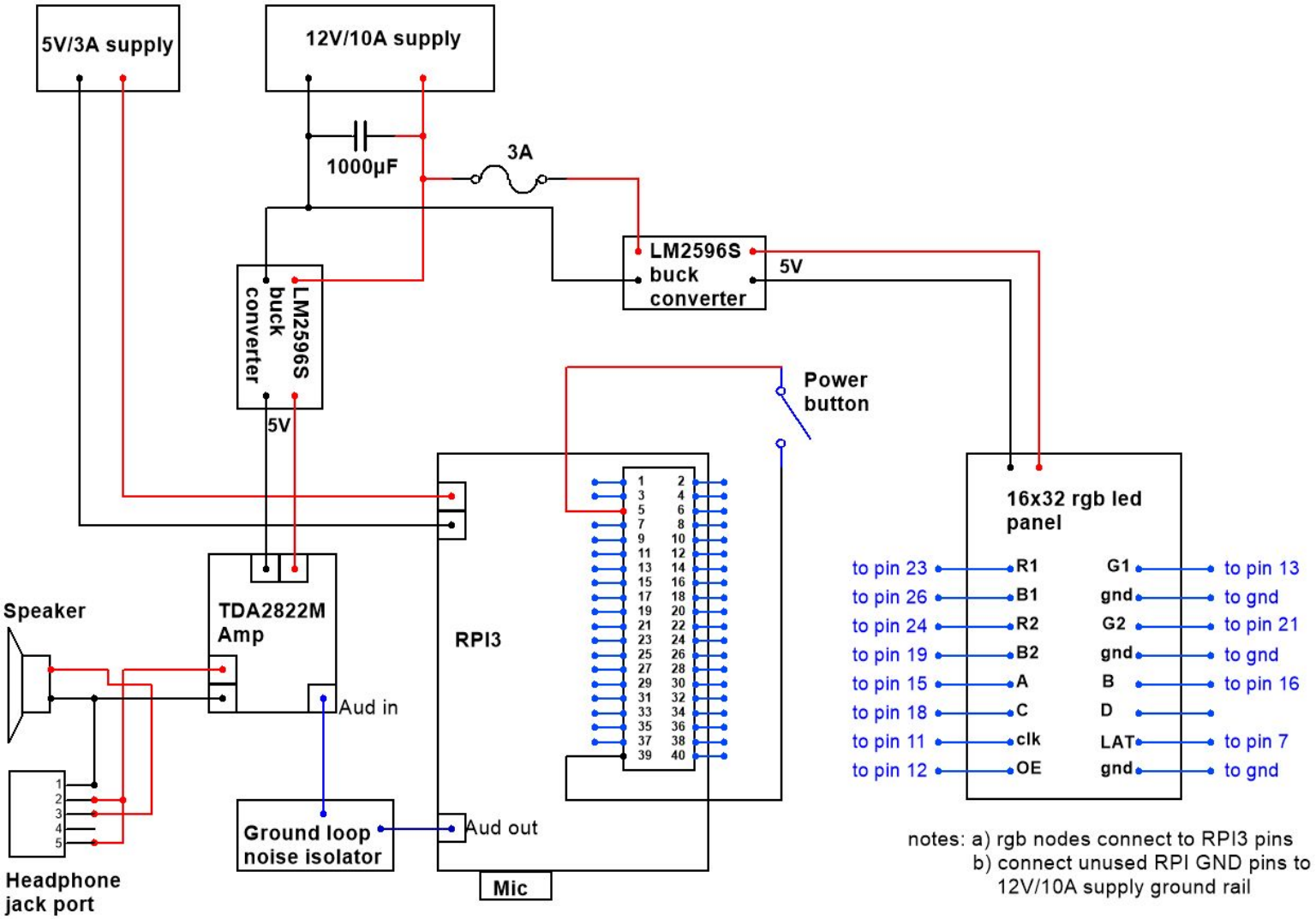


Figure 2.19: Complete Wiring diagram

ALEXA VOICE SERVICE

Through advances in technology many forms of communication have been created. People now have the means to communicate through sending emails, text messages, direct message on social media platforms, video calls, and “old school” phone calls talking.

Communication through speech is one of the first things we learn and it is also essential to building companionship. Therefore, this form of communication between the user and our companion is very important. As such we chose the Alexa Voice Service which allows the user to communicate through speech, the Alexa voice service will also allow the user access to the Alexa Cloud.

The Alexa Voice Service or AVS allows developers to access Amazon's Alexa cloud based capabilities. It works by capturing the users voice and sending it to the 'shared data stream' which then send the data to both the 'audio input processor' and the 'wake word engine.' The wake word engine detects if a wake word was spoken and then triggers the 'audio input processor' which handles the audio and sends it to the 'Alexa communication library.'

This creates a continuous connection between our device and the cloud which handles the Alexa driven interactions. The 'capability agent' correspond to a specific interface made visible by AVS API, which include speech recognition and synthesis, timers and alarms, notifications, weather reports, and thousands of custom skills.

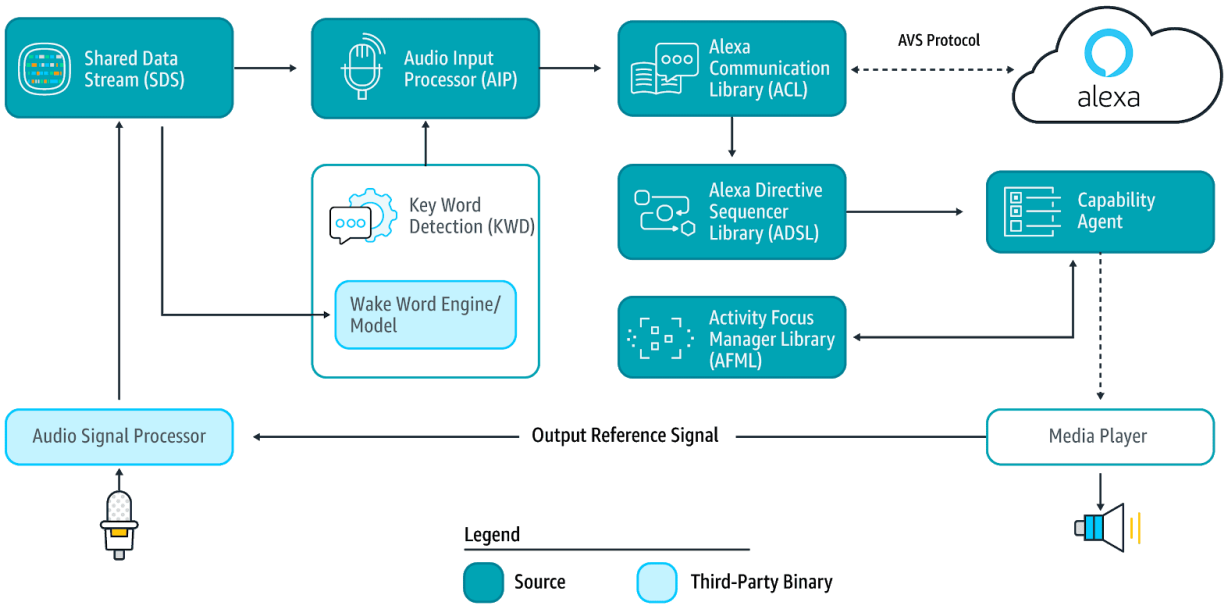


Figure 3.1: Alexa Voice Service Structure

To connect our device to the AVS we must go through the following procedure:

1. We created a developer account with amazon at developer.amazon.com.
2. Once the account is created, we selected alexa voice service and clicked on get started
3. We then filled out the product information section and the LWA security profile
4. Recorded the Security Profile ID, Client ID, and Client secret for later use
5. Opened the link https://developer.amazon.com/lwa/sp/overview.html and confirmed the Security profile
6. Added the url http://example.com to the Consent Privacy Notice URL section and then clicked save

Once those steps are complete the AVS can be installed on the Raspberry Pi [6].

1. We cloned the github repository with the command: cd ~/Desktop && git clone https://github.com/alexa/alexa-avs-sample-app.git
2. To update the Product ID, Client ID, and Client Secret to the correct values we previously recorded we ran the command: cd ~/Desktop/alexa-avs-sample-app && sudo nano automated_install.sh and saved those changes
3. Next we ran the install script: cd ~/Desktop/alexa-avs-sample-app && . automated_install.sh
4. A couple of question are asked and we answered y,y.y,1,1.y the installation took around 1 hour
5. We created a custom wake word using the https://snowboy.kitt.ai site and changed it by running the command: cp Nala.pmdl /home/pi/Desktop/alexa-avs-sample-app/samples/wakeWordAgent/ext/resources/alexa.umdl
6. To get the microphone working we ran: sudo apt-get install python-pyaudio python3-pyaudio sox
7. We edited the .asoundrc file so that the playback and recording devices corresponded to the correct cards and devices (.asoundrc Code in Appendix A)
8. We then ran our “initial_nala_startup.sh”, to run the web service and for authorization, as well as the sample app so that we can communicate with the AVS, and to also run the wake work engine snowboy so that we can interact with our companion using our custom wake word “Nala”
9. Lastly we created the script called “nala_startup.sh” which adds the LED matrix script to the initial startup and added the path /home/pi/nala_startup.sh to the autostart section of the raspberry pi so it that is would autostart. Using the command: cd .config/lxsession/LXDE-pi && sudo nano autostart

AMAZON WEB SERVICE

Amazon Web Service or AWS is a service that Amazon provides on demand cloud computing platforms for user to use on a paid subscription basis. The Alexa Skill Kit is a collection of self-service APIs and tools that makes it easy for you to create new voice-driven capabilities for Alexa. The code for the Skills are uploaded to AWS lambda which then executes the code. These custom skills are triggered by specific activation phrases to call upon specific functions.

AWS lambda is the computing service that will help our group run code without managing servers. One feature of AWS lambda is that you only have to pay for the time spent

computing. Currently we have a one year free subscription but once it expires normal fees will be applied which include monthly request charges and monthly compute charges. The total charges depends on the usage of your functions. There are many different rates which depends on the amount of memory allocated to your functions. For this project we created a skill that will state facts about New York.

To create the custom skill “New York Facts” that can be utilized by our device we must go through the following procedure:

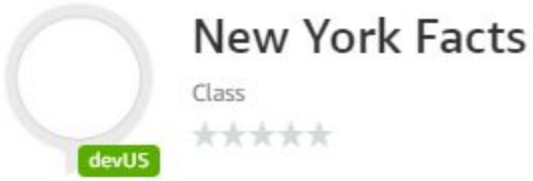
- Create an AWS account with the link aws.amazon.com
- To complete the account a credit card is needed, however we will be using the AWS free tier
- Creating the custom skill New York Facts based on the fact skill blueprint from amazon
- Open the amazon develop site and navigate to the alexa skill kit and click on create a skill
- Name the skill “New York Facts” and select the custom model
- We started with creating the invocation name to be the same as the skills name
- We then created the intent GetNewFactIntent and sample utterances which are way a user might invoke the intent (InteractionModel_NewYorkFacts Code in Appendix A)
- We also added 3 built in intent RepeatIntent, YesIntent, and NoIntent
- Then the model was saved and then built
- A new tab was opened to https://aws.amazon.com/ and we logged in and select lambda
- Click create function and select blueprints and search for facts and choose alexa-skill-kit-sdk-factskill
- For the NAME we used the same name as the skill “NewYorkFacts” and created a custom role which provides the permissions for the function to run
- Once the role is selected we clicked create function
- We added the trigger Alexa Skills Kit and modified the provided code to suit our needs (index_NewYorkFacts Code in Appendix A) and saved the changes.
- Copy the ARN and go back to the developer.amazon.com and New York Facts tab and paste it in the endpoint’s AWS Lambda ARN section.
- We then clicked save model and build model again, to complete building the Skill.
- The New York Facts is now ready to be called upon.

CAPABILITIES

By using the Alexa Skills Kit and AWS lambda we are able to create functions that can entertain the user and also create functions to improve their quality of life. Amazon allows you to publish those skills so that they can be accessed by users around the world. There are many type of skills that can be created from Smart Home Skills which allow users to control their smart appliances, skills which allow users to listen to the news, play games and record reminders. We used skills from the alexa skills library [7] which are listed below. However we also created a custom skill to tell us facts about New York but, skills are only limited by the creator's imagination. Amazon has a library of skills that exceeds 25,000 skills and the number of skills is only growing.

To activate a skill the user must say "Nala, Open <Skill Name>"
To Quit any skill the user must say "Nala, Stop"

New York Facts



To activate a skill the user must say "Nala, Open New York Facts"

ABOUT THIS SKILL
This skill contains over 25 interesting facts about New York and will randomly select one of these facts to tell you about.
If you need to hear a fact again say "Nala" followed by any of the following phrases:
"start over",
"can you say that again",
"can you start again",
"repeat that please",
"say that again",
"what was that",
"what did you say",
"can you repeat that"

Memory Challenge

ABOUT THIS SKILL
Memory challenge tests your memory skills by giving you five random words which you have to memorise and then asks you some questions whose answers lie among the words that were given by Alexa in the beginning of the game.

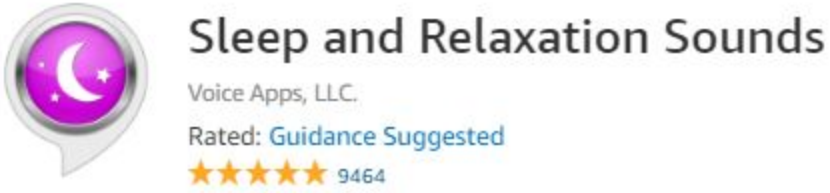
During the course of the game, you can use the following phrases while playing:
1. To repeat the words given in the beginning, just say 'Can you repeat the words?'
2. To listen to the question again, you can say 'repeat the question'
3. If you don't know the answer, you can pass the question by saying: 'I want to pass this question' or 'I don't know'
4. To start over the game, just say "start again" or "restart"

Ambient Sounds: Rain Sounds

ABOUT THIS SKILL
Rain Sounds is an ambient sound loop to help you sleep, relax, meditate, relieve stress, or block out unwanted noises.

By default the sound will play for 1 hour. To loop the sound until you say "Alexa, stop", just say "Alexa, loop on" while the sound is playing. The Skill will remember your preferred looping setting for future uses so you don’t have to say it again!
Please leave a 5-star review if you love this Skill, or send us an email with your suggestions at support@invokedapps.com. Your 5-star reviews encourage us to keep making more great ambient sounds for sleep and relaxation!

Sleep and Relaxation Sounds



ABOUT THIS SKILL
Sleep Sounds lets you play ambient sounds to help you sleep peacefully or block out unwanted noise at work or home. With over 50 high quality sounds to choose from, you'll be able to sleep better, stay focused, study without interruption, relax quickly, meditate more effectively, and get your baby to go to sleep faster! If there's a sound you'd like to see added, just e-mail us at support@sleepsounds.io.

HOW TO USE:

To Get Started: Say "Alexa open Sleep Sounds". Then, to play a sound say "Play Thunderstorm" or "Play Rain". To get a list of sounds, say "List Sounds". To have Sleep Sounds pick a random sound for you, say "Play Random".

When a sound is playing, you can say "Alexa Next" or Alexa Previous" to play the next or previous sound.

To start a sound directly (when Sleep Sounds is already playing a sound -OR- when Sleep Sounds is not started): Say "Alexa, tell Sleep Sounds to play Thunderstorm". Replace 'Thunderstorm' with any supported sound name.

By default, the sounds will loop automatically and play until you say "Alexa, Stop". To limit the time that the sound will play, first play the sound using one of the above methods and then say "Alexa, set a sleep timer for 2 hours" or whatever time limit you would like.

Ask My Buddy

To activate skill user must say “Nala, ask My Buddy to send help”

ABOUT THIS SKILL
Ask My Buddy helps you ask for assistance using only your voice - and - makes it easy to let your contacts know you are doing ok.

Alert feature
You can't predict when you'll need help, Ask My Buddy lets you immediately alert someone in your Personal Alert Network that you need them to check on you. You may alert just 1 contact or all of your friends or family with just one command. Simply say, 'Alexa, Ask My Buddy to alert YourContact' or 'Alexa, Ask My Buddy to alert everyone', and Ask My Buddy will immediately send an alert, notifying them to check on you right away. Alerts will be sent by text message / SMS, an email, and even a voice telephone call.

The Daily from The New York Times

To activate skill user must say “Nala, what’s in the news”

ABOUT THIS SKILL
This is how the news should sound. Fifteen minutes a day, five days a week. Only what you want to know and none of what you don’t. Hosted by Michael Barbaro. Powered by New York Times journalism. Ready by 6 a.m.

Short Bedtime Story

To activate skill user must say “Nala, launch Bedtime Story”

ABOUT THIS SKILL
Have Alexa tell a short bedtime story, personalized to your name.

There are several stories at this time and more are coming.

If you want to fix issues with names like if Bedtime thinks "Taylor" is a girl instead of a boy, or if Bedtime mispronounces the name like "Allie" comes out "Ellie", you can use the companion website http://bedtime.webguild.com. First get a Linking Code by asking, "Alexa, ask Bedtime Story to Configure". Have a notepad handy to write down the info.

Also at the Bedtime website, you will be able to disable any stories you don't want to hear and you will be able to author your own stories tailored to your family.

Enjoy!

Escape the Room

To activate skill user must say “Nala, open Escape the Room”

ABOUT THIS SKILL
***UPDATE: We've added a new room! Check out and try to escape from our new room, the Garage. Stay tuned for more!

If you want to be the first to hear when we add new rooms and new games please add yourself to our email list - https://goo.gl/forms/4oHEEGweuyzfBspV2.
Stay tuned - big things are coming!
***

You are trapped in a room. It's up to you and only you to escape. Search your room, pick up items, and solve puzzles to find your way out. There are three rooms to choose from and more being added. The skill will track your stats and how fast you complete each room.

The rooms in relative order of difficulty are:
Jail Cell - Easiest
Office - Medium
Car - Hardest

To play the game you have three basic actions (plus some more contextual commands you'll find out in the game) Sticking to these commands is the key to having the game work smoothly:
Look {Direction}
Inspect/Look at/use {Object}
Use {Item} on {Object}

Remember --- To inspect/look at/use an object, or use an item on an object, you must first be looking in the direction where you saw that object first.

Other useful commands are:
Help - can remind you what you can do at that point in time
Inventory - reminds you what items you have
Stats - will let you know how many times and how fast you completed each room

Once you master the commands it is important to note that once alexa starts speaking, you can just say, "Alexa", and then your next command, without waiting for her to finish speaking. This is very useful when you are exploring through parts of the room that you have seen already and don't need to wait for the full descriptions again

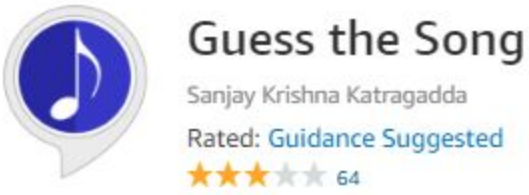
If you happen to get stuck, don't worry. It is a pretty hard game to solve. Think about all the clues you have and all the items you have picked up. If you still can't seem to solve the rooms don't be discouraged. We offer walkthroughs for each of the rooms which you can refer to help you escape!

Walkthroughs:

FREE -Jail Cell: https://www.stokedskills.com/jailcell.html
FREE - Office: https://www.stokedskills.com/office.html
FREE - Car: https://www.stokedskills.com/car.html

Please leave a nice review if you like the game. We really appreciate it and it help encourage us to keep working on great free skills for you. Any questions or support can be emailed to support@stokedskills.com. If you get stuck then please take a look at the walkthroughs so you can play through the game before leaving a 1* review saying the game was too hard.

Guess the Song



ABOUT THIS SKILL
It's easy to guess a song when you have music. But can you guess the same song by just listening to the lyrics?

We have a fun skill for you.
• Alexa reads a few lines from a popular song and you have to guess the name of the song.
• You can choose a playlist between 70s and 2010s.
• Each game consists of five rounds. You gain a point for every correct answer.
• You may also ask for a hint at any time.

Have fun :)

Millionaire Quiz Game

ABOUT THIS SKILL
Millionaire Quiz Game is an exciting game in which the more questions you answer, the more money you win. Do you have what it takes to be the next Millionaire?

HOW TO PLAY:
You'll be asked a series of 14 questions. With each question, you'll be given four possible answers and will be asked to choose the correct one. If you get it right, you'll win money and move to the next question worth even more money. If you get all 14 questions correct, you'll win $1,000,000.

If you get into trouble, you can use up to three lifelines. 50/50 will remove two of the wrong answers leaving you with the correct answer and one wrong answer. You can ask the question on social media and find out what others think the answer is. You can also phone one of your friends and have them help you answer the question.

With over 2500 questions and more being added each week, Millionaire Quiz Game will bring you and your family hours and hours of fun!

GETTING STARTED:
Just say "Alexa, start Millionaire Game" to get started!

DISCLAIMER:
This game is for entertainment only. No real money is won or lost.

FEEDBACK:
We hope you'll love Millionaire Quiz Game and give us a 5-star rating. If you feel that Millionaire Quiz Game deserves less than a 5-star rating, please let us know at millionairegame@voiceapps.io and we'll do everything in our power to make it better.

CONTACT US:
We'd love to hear from you! If you have questions, comments, or feedback, please e-mail us at millionairegame@voiceapps.io.

Any.do

ABOUT THIS SKILL

Get More Done with Any.do, the #1 To-do list, Calendar & Reminders App. Now works in perfect harmony with Amazon Alexa.

To get started, enable the Any.do Skill on Alexa, grant Lists read/write permissions and link to Any.do by creating a new account or signing-in to your existing account.

Once connected, your Alexa Shopping & To-do lists will be added as new lists to your Any.do account (at the moment, these are the only lists available on Alexa).
Then just say ‘Alexa, add birthday plans to my To-Do list’ or ‘Alexa, add tomatoes to my Shopping list’.
Your tasks will immediately sync to your Any.do account on mobile, desktop and tablet - so you’ll have it with you anytime, anywhere.

New to Any.do?
Visit https://www.any.do to download the app and start getting more done!
If you have any questions or issues connecting Alexa with Any.do, please contact customers@any.do

3D DESIGN

Design requirement:

Nala needs to be visually pleasant and cute, to allow users to feel comfortable with it. Nala should be physically durable and easy to transport. The device should be able to display emotion in response to users' input. This version of the prototype should be serviceable, should new parts need to be installed or if a future team decides to build it further.

Design background:

To make Nala pleasant and cute looking, we looked at existing robotic companion devices. Professor Xiao showed us a device on the market that is shaped like cat crossed with a stereo system. We took this as our initial visual design target. The idea was simple:

- The device would be durable plastic, and easy to move around like a small stereo
- The robot would convey its emotions through a large LED screen or display
- The robot would be shaped like a cat
- The robot would be able to "hear" the user through an internal microphone and respond through a speaker

Visual inspiration and initial sketches:

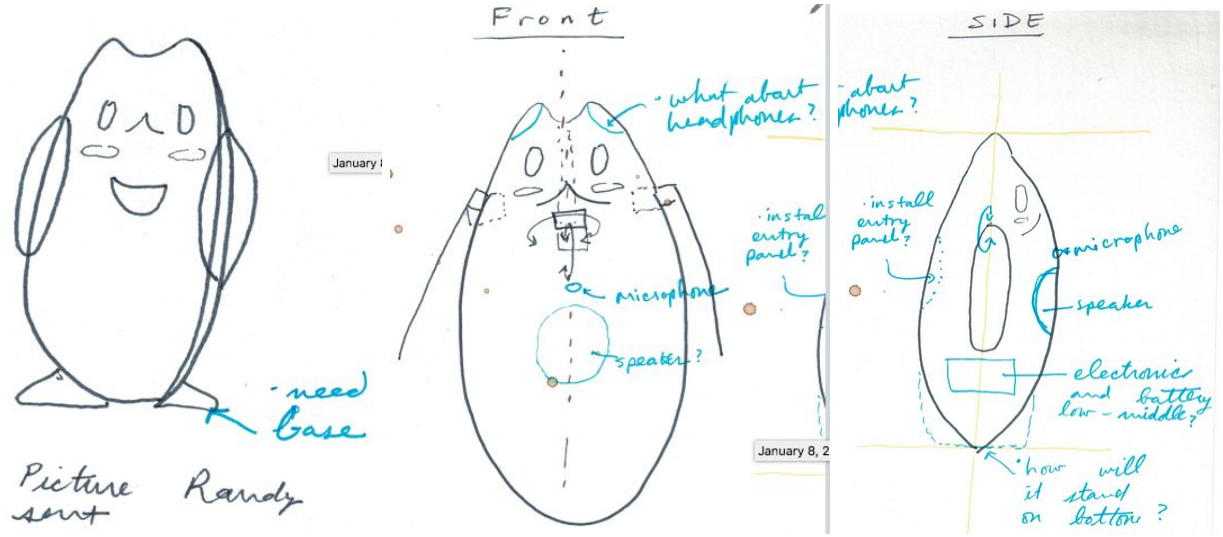
We found a cute image of a cartoon cat that became our original inspiration image, a cat that had simple lines and was undeniably cute (Fig. 4.1):

Fig. 4.1. Initial visual design inspiration.

We did quick sketches to envision how users would interact with Nala (Fig. 4.2).

Fig. 4.2. Sketches of how user would interact with Nala.

Through sketching, we started to uncover more details of the project's physical design. We made the choice that the robot would sit on a table or flat surface. This meant that the robot would need to have a flat, sturdy base. We also started to sketch the electronics components the robot would need, and brainstormed where they could go on the body.



Figs. 4.3. Locating electronics on Nala.

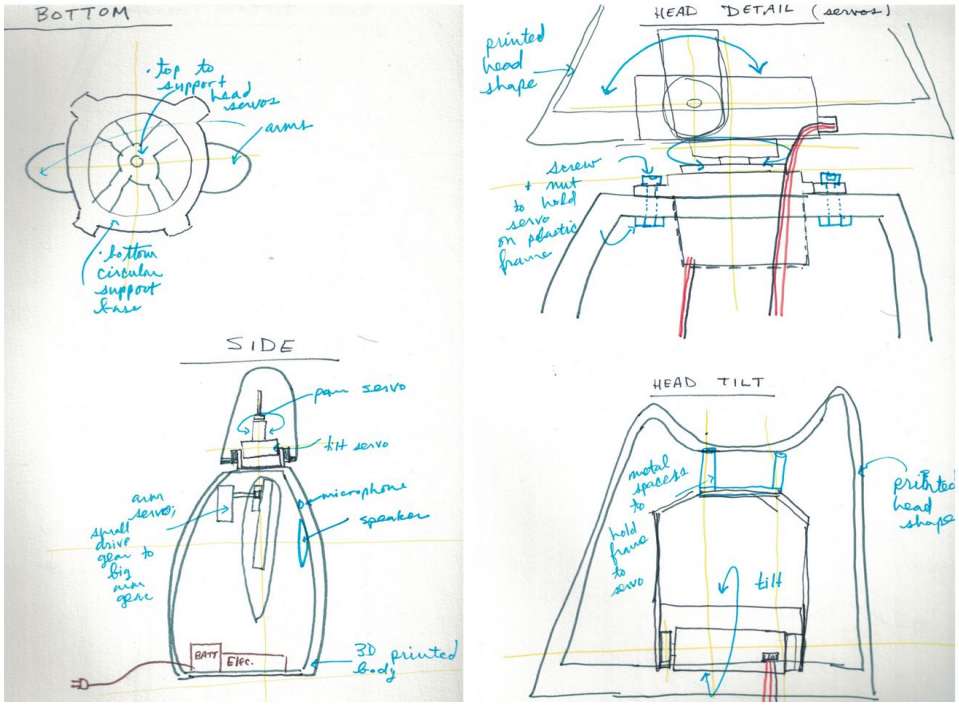


Fig. 4.4. Nala with arm and neck motors sketched.

As seen in Fig. 4.4, we also considered having a separate head and body for the robot. We also intended for Nala to have moving arms, which would allow Nala to show emotion. We planned to have four motors to allow arms to move up and down, and to have a neck that could swivel and pan up and down.

Final Design:

After meeting with our project mentor Professor Xiao, we decided that for the amount of time we had, we would have to simplify our robot's body. We would not be able to have moving parts, and we would simplify the robot to become just a single, sturdy unit: a cat's head. We began drawing a design in Fusion 360 to represent this:

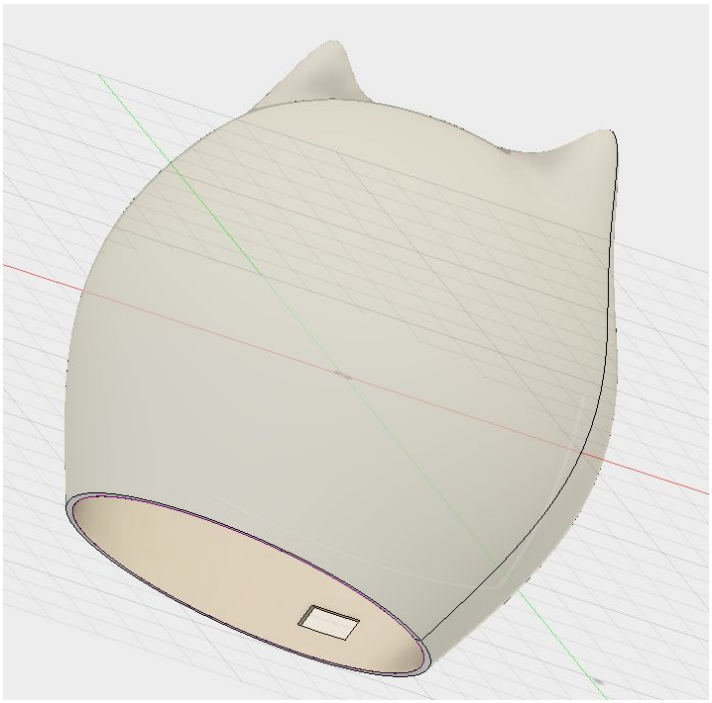


Fig. 4.5. Initial cat head design.

Soon after this, we created a new version which had more “catlike” features:

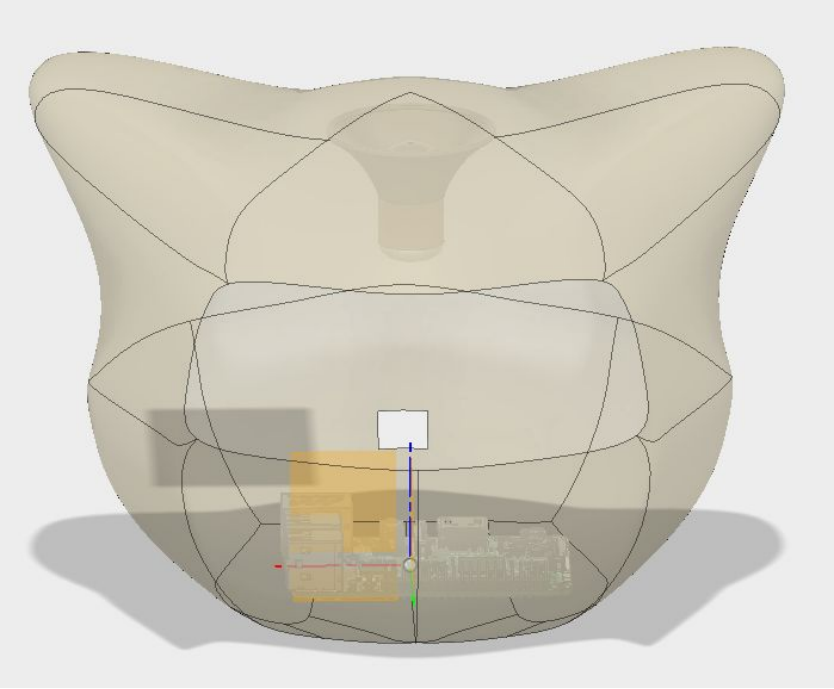


Fig. 4.6. Nala with more catlike ears.

However, to simplify the body geometry for splitting into several sections for printing, and later assembly, we decided to create a circular body, which would be cut in half. Also, we created even more catlike ears (Fig. 4.7).

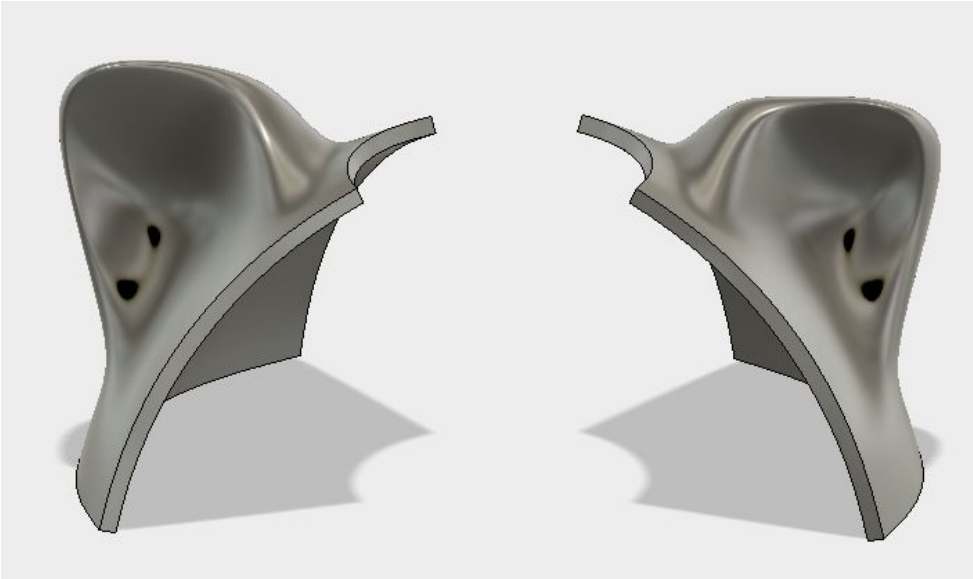


Fig. 4.7. Nala's final feline ears.

We then created a round body shape, that began from a "globe" shape, 10 inches in diameter. The top half of the globe became the basis for the head and face. It was slotted on the top so that a speaker could play audio out of it.

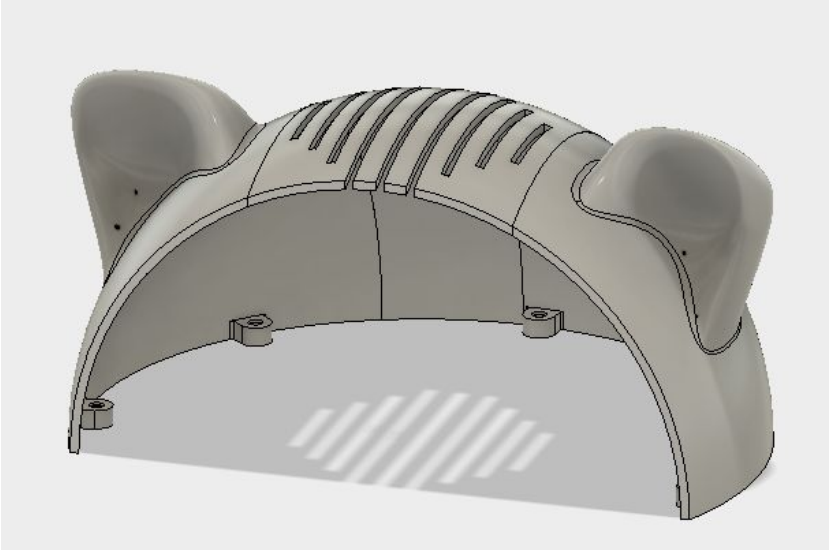


Fig. 4.8. Nala's head with speaker vents.

On the interior of Nala's head are reinforced M3 hardware copper nut holders, which have a 4 mm diameter hole going through the middle of the support, and are surrounded on each side by 4 mm of support material.

Fig. 4.9. Copper inserts to hold mounting bolts.

We needed to mount an LED display to Nala's font. We considered several approaches. Dr. Chang Yong suggested we create "goggles" to hold the LED display, which was reminiscent of a Chinese cartoon, "Inspector Black Cat."

Fig.4.10. Inspector Black Cat for LED display reference.

Fig. 4.11. Nala's LED display mounted on head.

Next, we needed a "main entry" to install all the robot's internal components. We decided to use a bottom panel, which could be easily removed and replaced, and would be attached to the bottom via M3 bolt hardware.

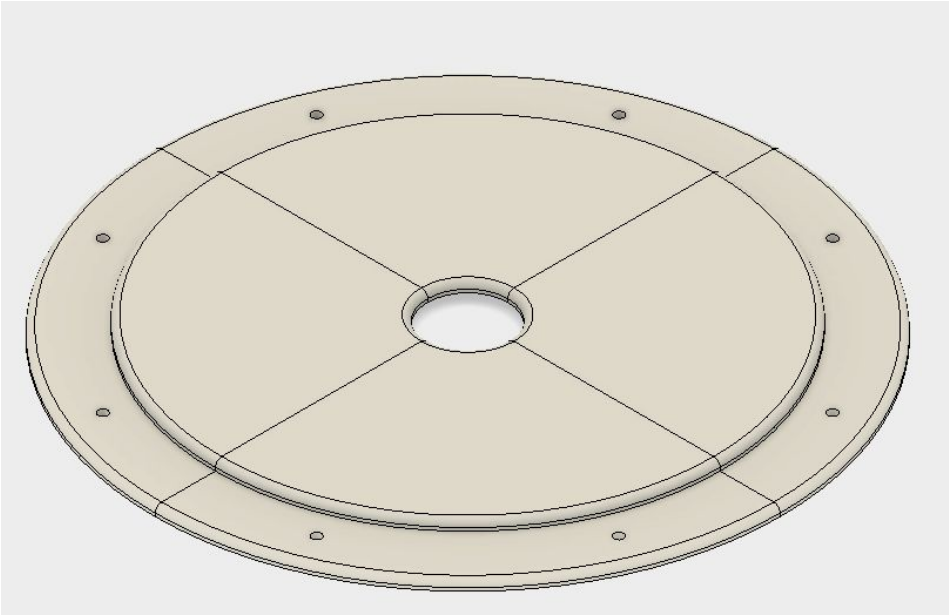


Fig. 4.12. Removable base plate for electronics installation.

The baseplate was cut into four sections so that it could be printed in our lab. The eight holes surrounding the outside edge are 3.5 mm each to allow an M3 bolt to pass upward through

them. There is a center hole large enough for a finger to fit through to pry open the compartment. The plate has two levels so that it fits with the contour of the body above it.

To allow the shelf to fit into the base, so that the bottom would be flush, we created an inner inset to match the shape of the base (Fig. 4.13).

Fig. 4.13. Cutaway of bottom showing base insert.

This insert allowed the base to fit easily into the bottom. Fig. 4.14. shows the complete revolution of the bottom base inset:

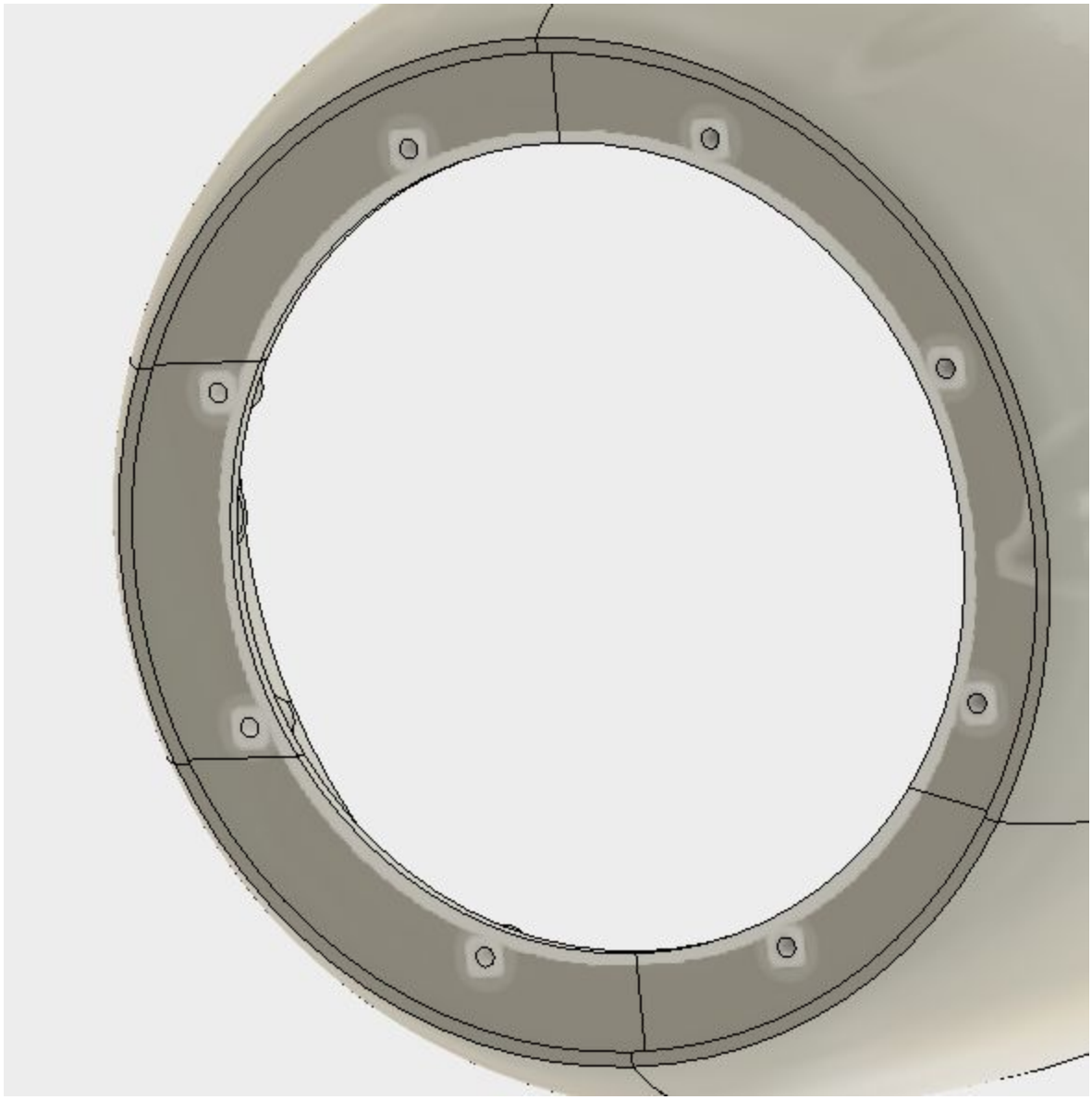


Fig. 4.14. Bottom base inset.

The complete bottom shell is 5” high and 10” in diameter (Fig. 4.15).

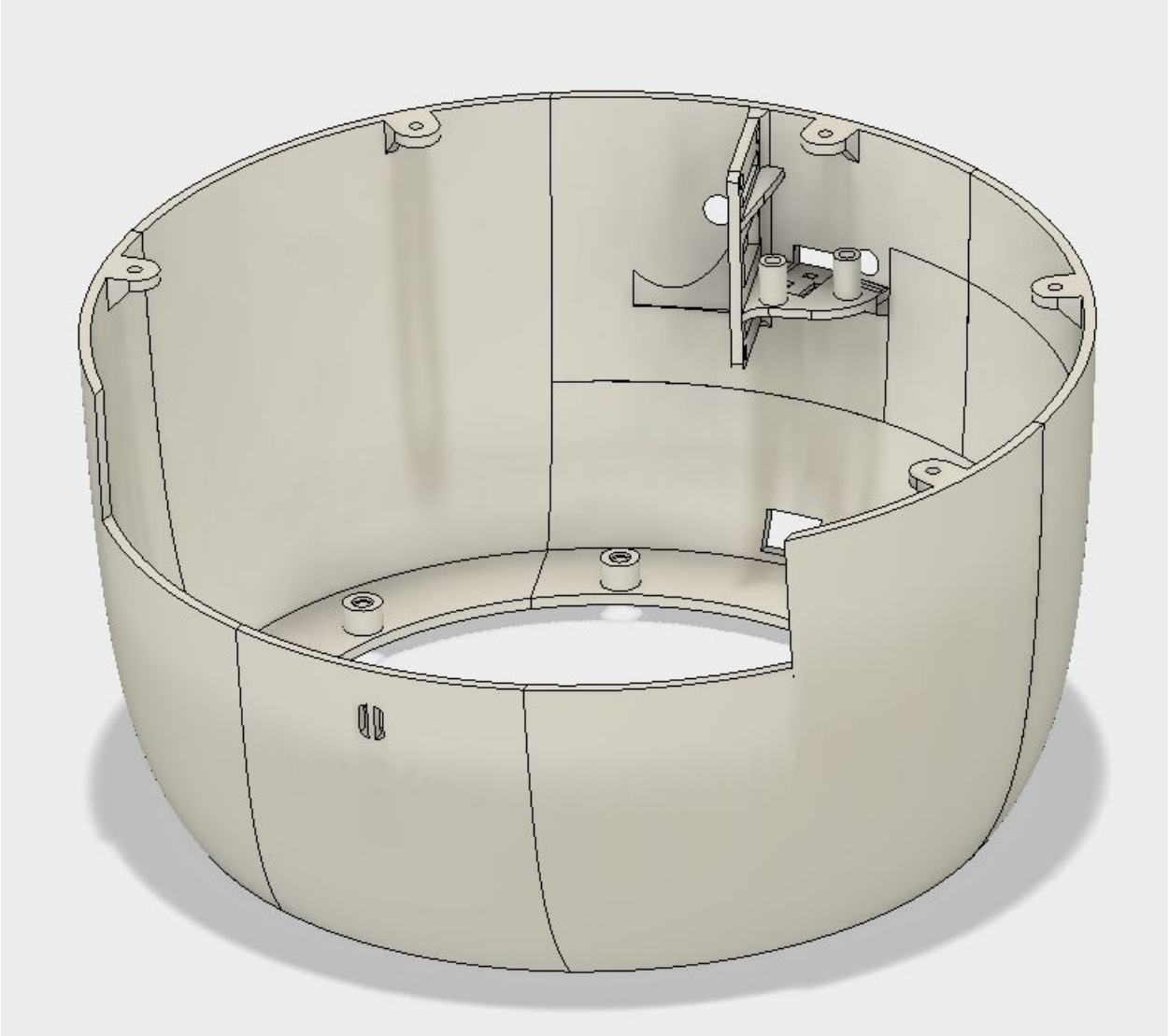


Fig. 4.15. Complete bottom shell.

At the front of the shell are two small slits which were intended for use with a circular eletret microphone:

Fig. 4.16. Electret microphone fitted for Nala's bottom shell

After printing the base bottom, we decided not to use this particular microphone. However, the bottom was designed to hold it with a ring printed on the inside of the front of the base:

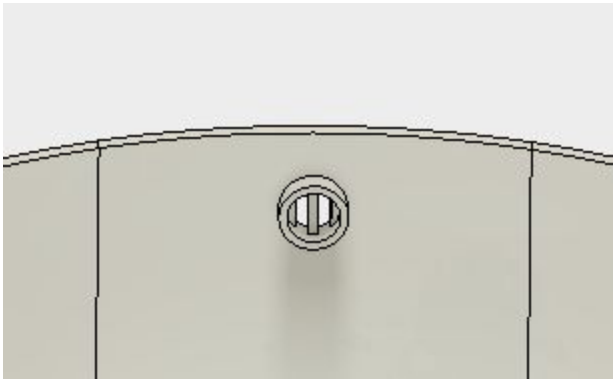


Fig. 4.17. Ring holder to hold elecrtet microphone.

A shelf was also created on the inside of the bottom base to hold electronic components. One particular challenge was how to mount rectangular printed circuit boards inside Nala's circular base.

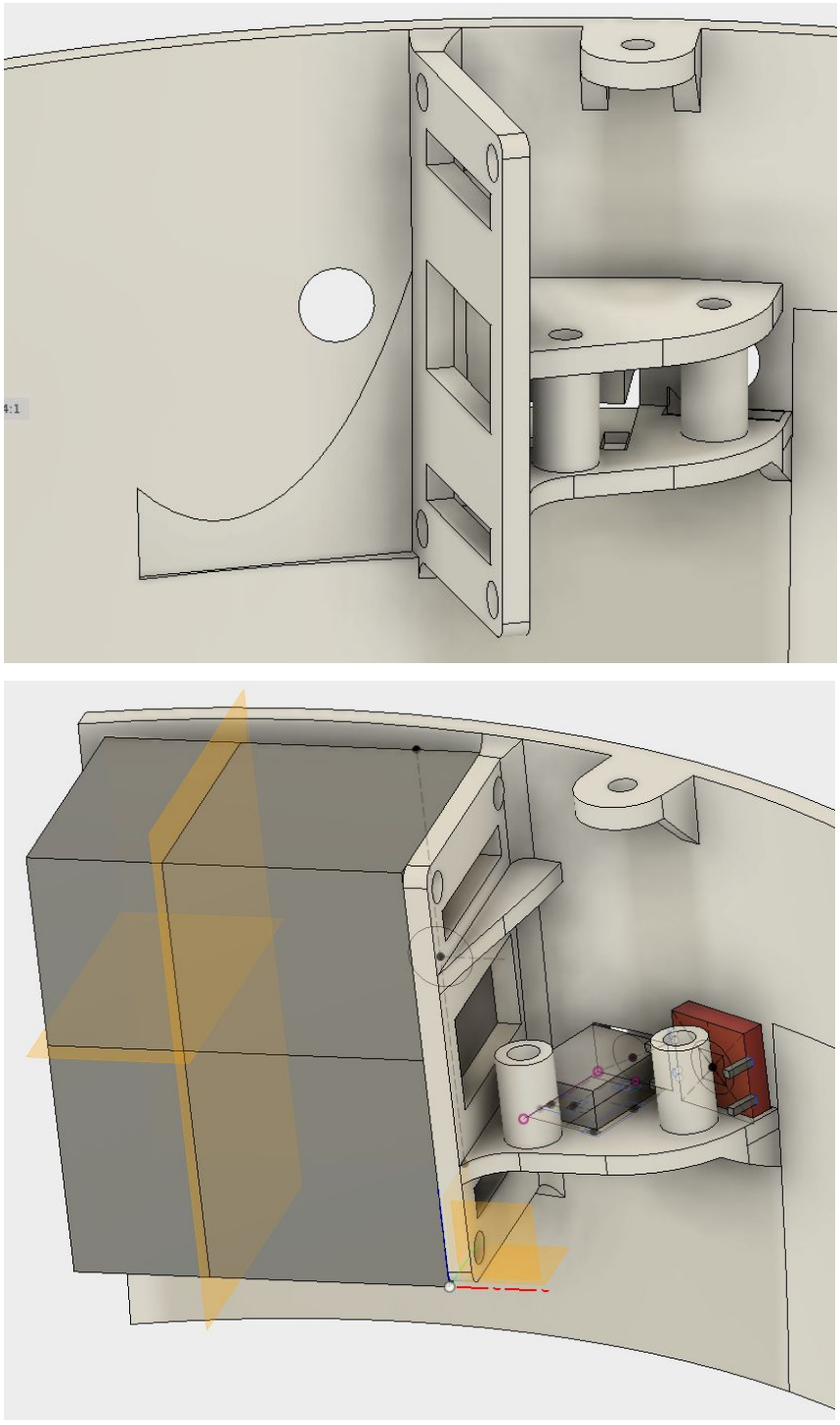


Fig. 4.18. Vertically-mounted electronics shelf with side shelf with separable top (top). Components added to show electronics installation with enclosing top removed (bottom).

We add ribs to the bottom of the electronics shelf so that the shelves are more sturdy. The audio jack and “on” button are inset into a platform, and then a top can be screwed down on top of them, to “squeeze” them into place.

The audio jack, audio potentiometer, and on button all have shafts that need to extend through the 3 mm body shell. After being installed behind the curved body boundary, it was challenging to make enough room for their shafts to clear the outer boundary of the body.

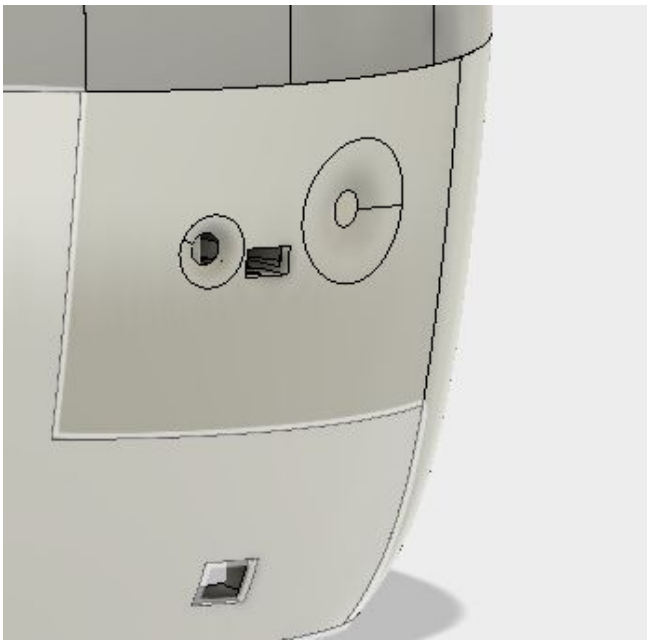


Fig. 4.19. Round fillets that allow for access to shafts passing through body

The final body needed to be cut up into several pieces so that it could be printed. We observed the size constraints per printed piece of 250 mm wide, 250 mm long, 250 mm high. The final bottom shell after being separated into different parts appears in Fig. 4.20.

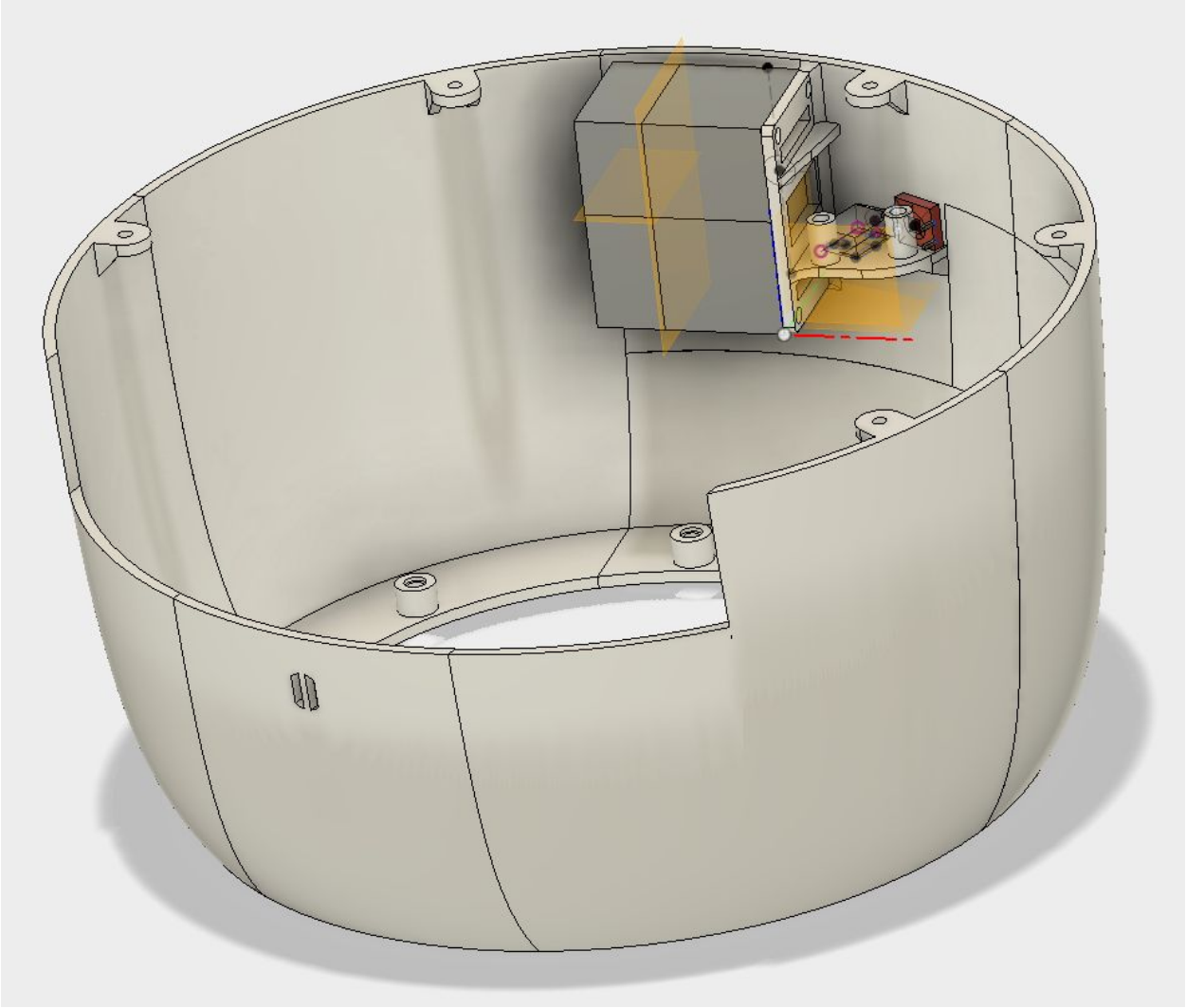


Fig. 4.20. Final bottom shell.

It was decided about halfway through the project that we would incorporate an LED display, to animate Nala's facial response and making Nala more responsive. The LED matrix chosen is described in the Electrical section. To mount this LED matrix, we chose to place it where Nala's "eyes" would be, on the top half of its body. A sturdy LED matrix holder was built:

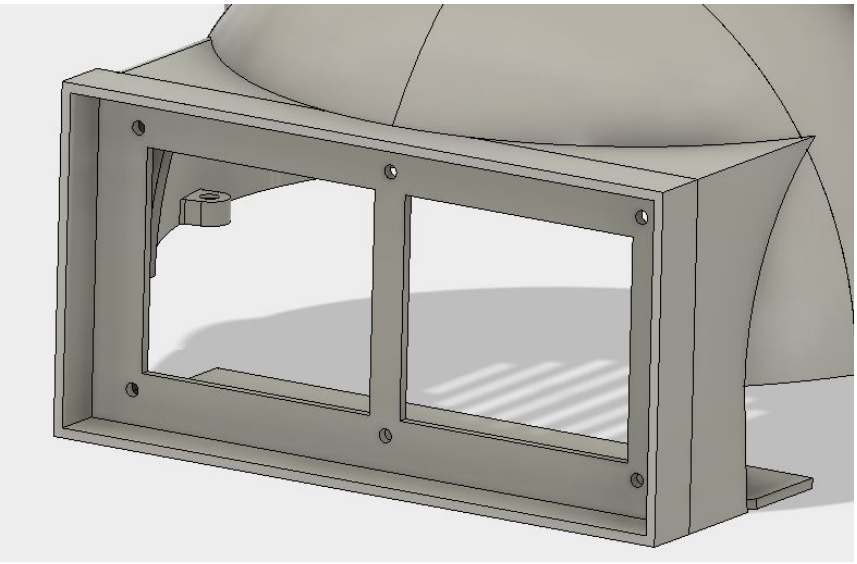


Fig. 4.21. Rectangular LED matrix holder.

This holder significantly changed Nala's look. The holder appeared as "futuristic" virtual reality goggles. We chose to embrace this as part of this design iteration. It diverges from the original vision of an organic, cute, animal look. Instead, it makes Nala look more futuristic and related to technology.

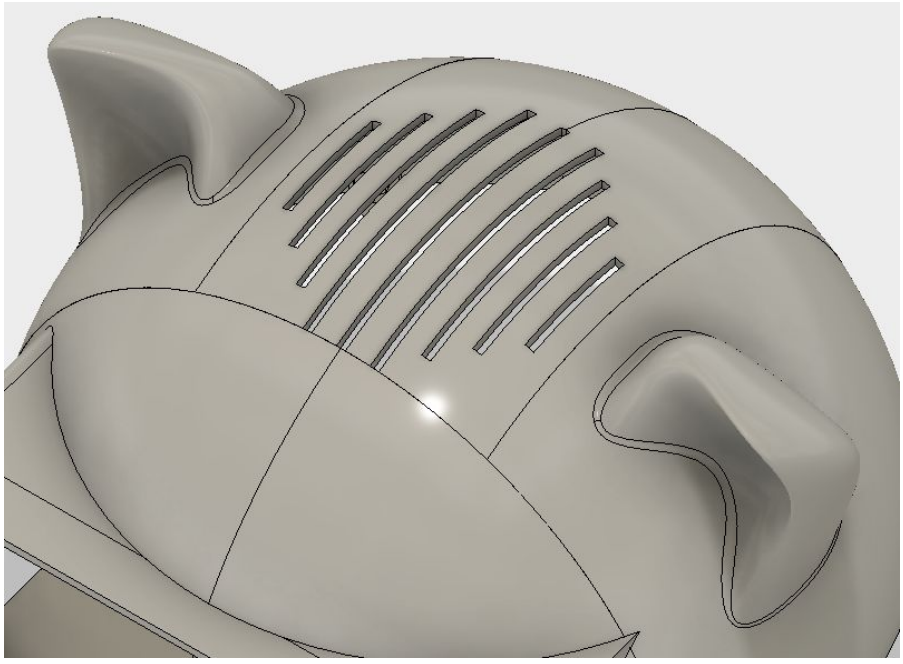


Fig. 4.22. Vented top to allow speaker access.

A speaker is installed in the top of Nala's top shell. A circular pattern of vents were created to allow sound to pass through the body and create a stylish look. In a future version, a

speaker holder would be installed beneath the vents. In this current prototype, the speaker is installed with screws that extend through the vent slits.

One of the most important visual features of Nala is the catlike ears. The ears point forward gently, like an affectionate cat.

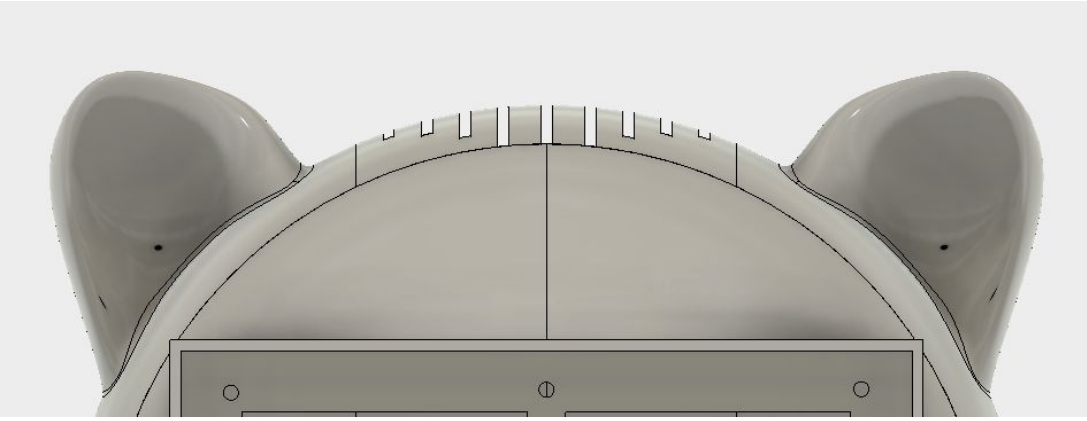


Fig. 4.23. Nala's catlike ears.

Inside the top shell are M3 brass nut holders, which allow the top and bottom shells to be reopened and closed mechanically, as opposed to permanently gluing them closed together. The nut holders are 4 mm thick on either side of the holder hole. This has created a very sturdy connection system.

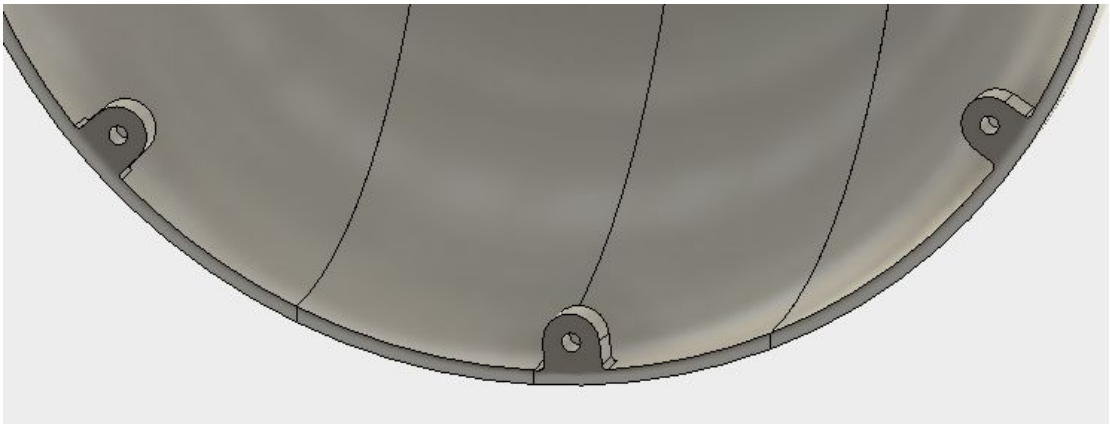


Fig. 4.24. M3 brass nut holders.

The complete top shell is 10 inches in diameter at the bottom, shown in Fig. 4.25. It was also cut into several pieces in the same manner as the bottom shell to allow the printer to print the separate pieces.

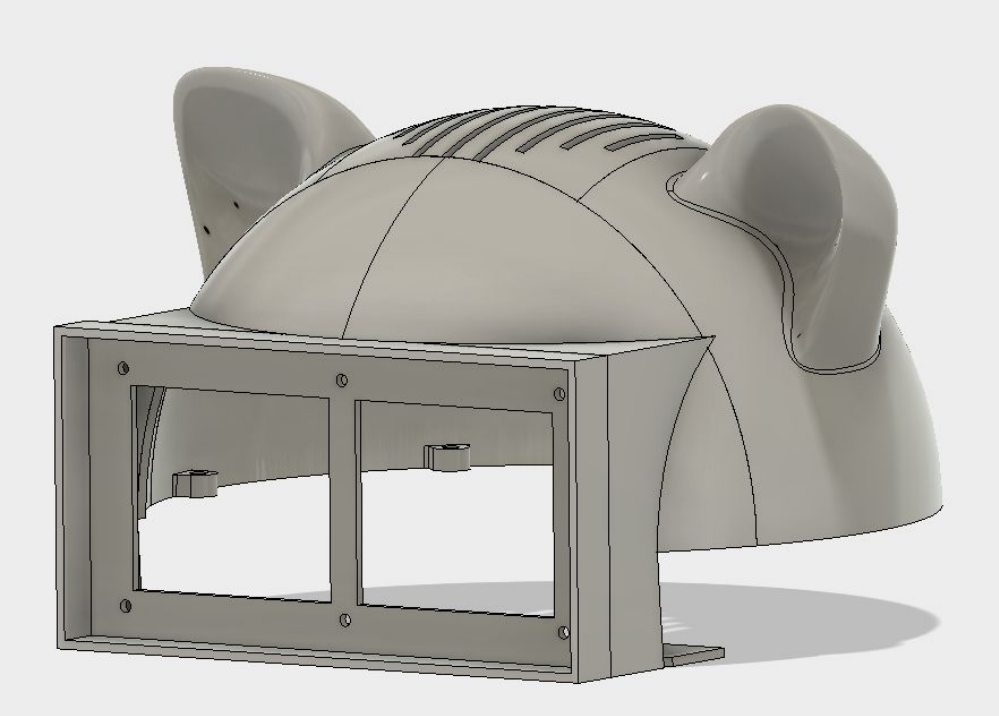


Fig. 4.25. Complete top shell.

We then are able to attach the baseplate, bottom shell, and top shell, completely with mechanical hardware, M3 bolts and brass nut inserts. The installation of all three pieces is shown in Fig. 4.26.

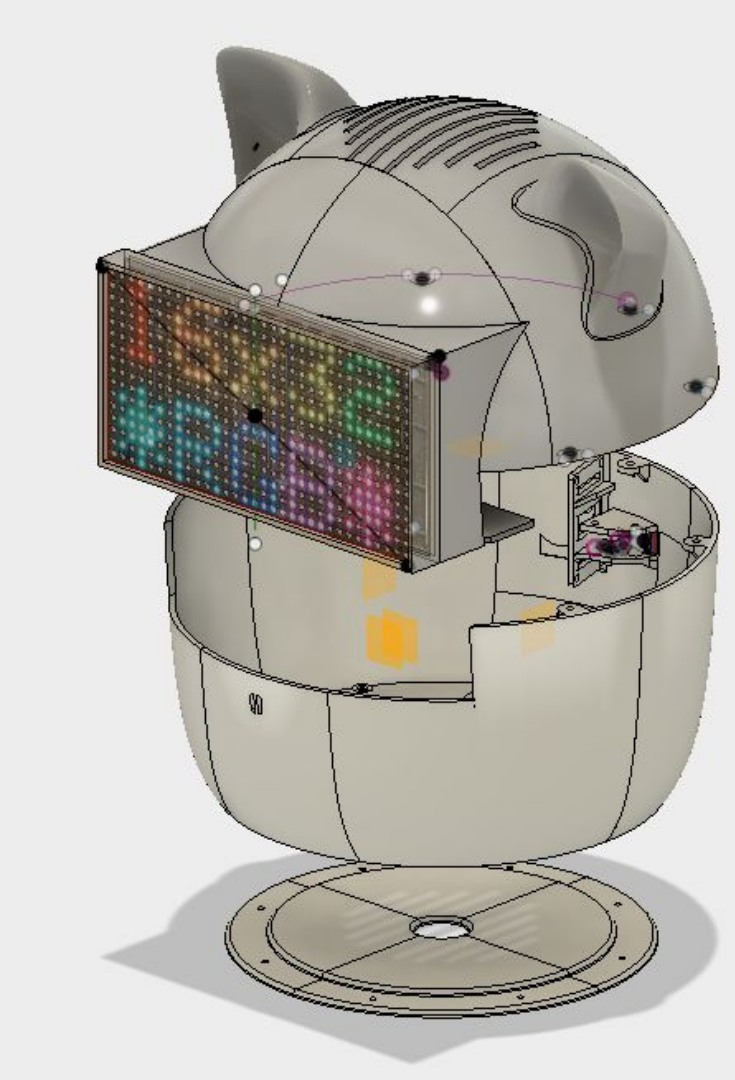


Fig. 4.26. Image of Nala's three main components.

Finally, the complete 3D design of Nala in Fusion 360 is shown in Fig. 4.27.

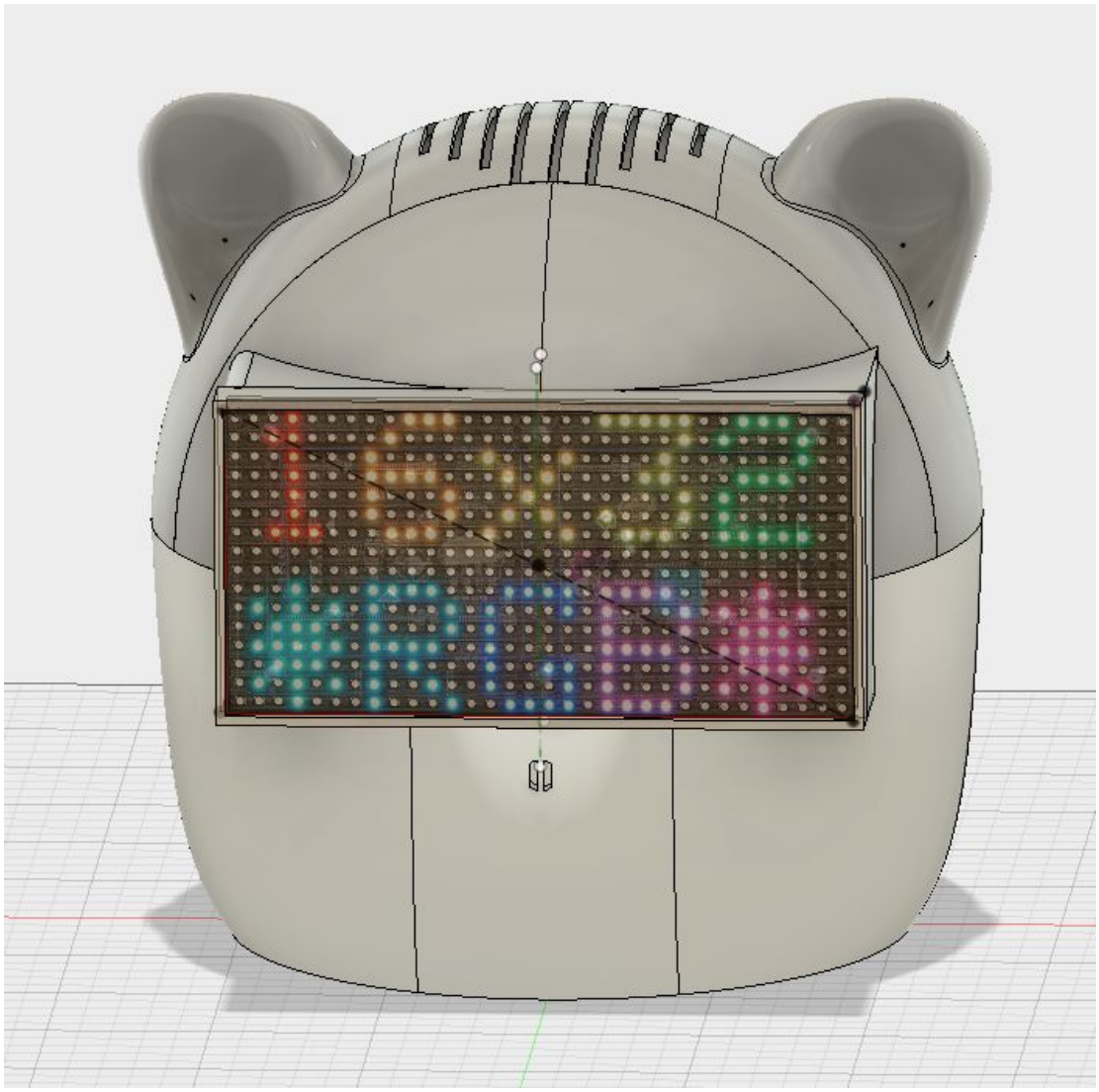


Fig. 4.27. Image of final assembly of Nala in Fusion 360.

Conclusion

After spending the first half of the year researching companion robots and the needs of senior citizens, we were able to decide on what our companion should look like and what it should be capable of so that it would be helpful. The goals of the project were achieved. We were able to create a cute, user-friendly, therapeutic, and interactive companion robot for the elderly. In a future phase of the project, the following improvements could be made. The first would be to incorporate a pulse band with Nala to monitor the user's vital signals, and sending necessary alerts in case of emergencies. Allowing Nala to control smart home devices such as lights, security systems, air conditioning units, and thermostats would be another added feature that would be very helpful with the increasing number of smart appliances. Lastly, by

incorporating motors to enable physical motion Nala could become more interactive. One of the best thing about Nala is the fact we can continue to add features to personalize it to a user's need. We learned a great deal through the journey of creating Nala about the Raspberry Pi capabilities, the functionality of the ever growing Amazon Alexa, and how much effort is put into designing objects through our use of Fusion 360. Furthermore, we also learned about the effort, work, and persistence that it takes in doing a group design project. This was a valuable learning experience that we will take with us into our future careers.

Appendix A: Amazon Alexa

Amazon

Alexa Skill Kit Code

InteractionModel_NewYorkFacts

```
{
  "interactionModel": {
    "languageModel": {
      "invocationName": "new york facts",
      "intents": [
        {
          "name": "AMAZON.CancelIntent",
          "samples": []
        },
        {
          "name": "AMAZON.HelpIntent",
          "samples": []
        },
        {
          "name": "AMAZON.StopIntent",
          "samples": []
        },
        {
          "name": "GetNewFactIntent",
          "slots": [
            {
              "name": "fact_type",
              "type": "fact_type"
            }
          ],
          "samples": [
            "{fact_type} fact",
            "i want to know about {fact_type}",
            "tell me a fact about {fact_type}",
            "tell me a {fact_type} fact",
            "a fact",
            "a new york fact",
            "tell me a fact",
            "tell me a new york fact",
```

```
            "give me a fact",
            "give me a new york fact",
            "tell me trivia",
            "tell me new york trivia",
            "give me trivia",
            "give me some new york trivia",
            "give me some information",
            "give me some information about new york",
            "tell me something about new york",
            "give me something"
         ]
      },
      {
         "name": "AMAZON.RepeatIntent",
         "samples": [
            "start over",
            "can you say that again",
            "can you start again",
            "repeat that please",
            "say that again",
            "what was that",
            "what did you say",
            "can you repeat that"
         ]
      },
      {
         "name": "AMAZON.YesIntent",
         "samples": []
      },
      {
         "name": "AMAZON.NoIntent",
         "samples": []
      }
   ],
   "types": [
      {
         "name": "fact_type",
         "values": [
            {
               "name": {
                  "value": "history",
                  "synonyms": [
                     "past",
                     "historical"
```

```
                ]
              }
            },
            {
              "name": {
                "value": "art",
                "synonyms": [
                  "paintings"
                ]
              }
            },
            {
              "name": {
                "value": "architecture",
                "synonyms": [
                  "buildings"
                ]
              }
            },
            {
              "name": {
                "value": "sports",
                "synonyms": [
                  "volleyball",
                  "hockey",
                  "soccer",
                  "football",
                  "baseball",
                  "basketball"
                ]
              }
            },
            {
              "name": {
                "value": "movie",
                "synonyms": [
                  "film"
                ]
              }
            }
          ]
        }
      ]
    }
```

```
    }
}
```

Amazon Web Service Code

index_NewYorkFacts.js

```
/* eslint-disable  func-names */
/* eslint quote-props: ["error", "consistent"]*/
/**
 * This sample demonstrates a simple skill built with the Amazon Alexa Skills
 * nodejs skill development kit.
 * This sample supports multiple lauguages. (en-US, en-GB, de-DE).
 * The Intent Schema, Custom Slots and Sample Utterances for this skill, as well
 * as testing instructions are located at https://github.com/alexa/skill-sample-nodejs-fact
 **/

'use strict';
const Alexa = require('alexa-sdk');

//=========================================================================
=========================================================================
====
//TODO: The items below this comment need your attention.
//=========================================================================
=========================================================================
====

//Replace with your app ID (OPTIONAL).  You can find this value at the top of your skill's
page on http://developer.amazon.com.
//Make sure to enclose your value in quotes, like this: const APP_ID =
'amzn1.ask.skill.bb4045e6-b3e8-4133-b650-72923c5980f1';
const APP_ID = 'amzn1.ask.skill.ad138521-682f-4da1-90a3-ac6d42749e14';

const SKILL_NAME = 'New York Facts';
const GET_FACT_MESSAGE = "Here's your fact: ";
const HELP_MESSAGE = 'You can say tell me a New York fact, or, you can say exit... What
can I help you with?';
const HELP_REPROMPT = 'What can I help you with?';
const STOP_MESSAGE = 'Goodbye!';

//=========================================================================
=========================================================================
====
```

```
//TODO: Replace this data with your own.  You can find translations of this data at
http://github.com/alexa/skill-sample-node-js-fact/data
//=========================================================================
=========================================================================
====
const data = [
    'New York is the third most populous state in the United States after California and Texas.
Of its population of 19 million, a little over 8 million live in New York City alone. In fact,
about 1 in every 38 people in the U.S. lives in New York City, and more people live in New
York City than in Australia and Switzerland combined.',
'While New York City is the largest city in New York, the state capital is actually Albany,
which has 1/80 the population of New York City.',
'Albany, the state capital of New York, is known as the city of many names. The original
inhabitants, the Mohicans, called the city PempotowwathutMuhhcanneuw, or "the 1st fireplace
of the Mohican nation," while the Dutch called it Beverwijck or "Beaver District." In 1664, the
English would name the city after the Duke of Albany.',
'The New York capitol building in Albany took 28 years (1867–1899), 5 different architects,
and over $25 million to build. Composed of granite, the building has 5 floors total and a
Million Dollar Staircase with 300 carved stone portraits of famous New Yorkers and others.',
'New York City, New York, is the most linguistically diverse city with over 800 languages
spoken, and 4 in 10 households speak a language other than English.',
    'The Woodstock Music Festival was actually held on a dairy farm in Bethel, NY, after the
towns of Woodstock and Wallkill refused to host it. The festival took place for three days in
August 1969 with 400,000 people attending.',
'Lake Placid, NY, is the only U.S. city to have hosted the Winter Olympics twice, first in 1932
and again in 1980.',
'According to a report in November 2015, New York is the second safest state in the U.S. to
drive in after Massachusetts, with 6.1 deaths per 100,000 state residents; 1,199
automobile-related deaths in 2013; and 91% of New Yorkers choosing to wear their seatbelts,
which is above the national average of 87%.',
'In 1848, the first womens rights convention in the United States was held in Seneca Falls, NY,
officially starting off the fight for womens rights. Women would win the right to vote in New
York in 1917.',
'A 2015 study found that New York is the 13th best state to live in, with one of the factors
being an average income of $58,878 per year, which is the 16th highest in the United States.',
'New York was the 6th state to legalize same-sex marriage when Governor Cuomo passed the
Marriage Equality Act on June 4, 2011. After the act was passed, 823 same-sex couples were
married in New York City alone.',
'Due to struggles with the potato famine and political issues, by 1850 more native-born Irish
lived in New York City, NY, than in Dublin, Ireland. New York City is still home to more
people of Irish ancestry than Irelands capital city.',
'New Yorkers bite 10 times more people than sharks do worldwide each year.',
'With 54,556 square feet, New York is the 27th largest state in the United States. The state of
Rhode Island would fit into New York more than 35 times.',
```

'Adirondack Park in northeast New York has 6 million acres, making it the largest national park in the United States. It is larger than Yellowstone, Glacier, Everglades, and Grand Canyon National Parks combined.',
'Niagara Falls State Park was the first state park in the United States when New York made it the Niagara Reservation in 1885. The three falls cascade 150,000 gallons of water over 176 feet and attract 12 million visitors each year.',
'The Niagara River has two hydroelectric plants that are capable of producing 2.5 million kilowatts of energy, which is enough to power the state of New York and the province of Ontario, Canada.',
'The worlds smallest church is found in Oneida, NY. The Cross Island Chapel is 51" x 81", making it large enough to sit two people.',
'With 18,000 cattle and calf farms, New York state is the third leading producer of dairy products in the United States.',
'From 1886 to 1924, over 14 million immigrants entered through New York harbor into the United States. About 40% of Americans can trace at least one ancestor to Ellis Island.',
'In 1901, New York was the first state to require all automobiles to have license plates. However, the plates were not issued by the state but were made by the owner and were required to have the owners initials.',
'The state of New York has produced more governors than any other state in the U.S. With 116 governors claiming New York as their birth state, Virginia is a distant second with only 78.',
'In 1971, the bloodiest prison rebellion to ever take place in the United States happened at the Attica State Correctional Facility in Attica, New York. A total of 43 inmates and officers lost their lives during the riot.',
'Oysters were such a popular food item in New York in the 19th century that the shells were actually used to pave Pearl Street in New York City. They were also used for lime for the masonry of the Trinity Church.',
'New York is known as the "Empire State" due to its growth and prosperity early in its history. George Washington is said to have seen New York as "the seat of the empire."',
'The Genesee River of New York is one of only 33 rivers in the world that runs from south to north.',
'The September 11, 2001, terrorist attacks on the World Trade Center in New York City were the worst single foreign assault on American soil. The attack was the nations deadliest tragedy for U.S. firefighters, and 2,753 people in total lost their lives.',
'In 2012, Hurricane Sandy took the lives of 150 people along the Atlantic Coast, about 50 of which were New Yorkers. Large portions of New York City lost electricity, 650,000 homes and businesses were lost, and the total damages were about $68 billion, with $33 million of the damage in New York state.',
'If the state of New York were its own country, its economy would rank 15th in the world, being slightly smaller than the economies of Canada and Spain. New Yorks economy ranks 3rd in the United States, after California and Texas.',
'France gifted the Statue of Liberty to the United States in 1886 as a celebration of 100 years of U.S. independence and continued allegiance between the two countries. The statue was shipped as 350 pieces in 214 crates and took 4 months to assemble at its current home on New York’s Ellis Island.',

'The New York Stock Exchange is the largest exchange in the world, with a trading volume of $5.5 million. The second largest is the NASDAQ, which is also located in New York.',
"New York City's Federal Reserve Bank has the largest gold storage in the world. The vault is 80 feet below street level and contains $90 billion in gold.",
'The state of New York disposes of nearly 23 million tons of trash each year. New York city alone sends a 9-mile-long fleet of trucks of trash each day to sites up to 300 miles away.',
'Chittenango, NY, is the home of The Wizard of Oz author L. Frank Baum. The town has yellow brick sidewalks that lead up to The Wizard of Oz themed businesses, such as Auntie Ems, as well as an annual Munchkins parade.',
'The New York Yankees have won baseball's World Series 27 times as of 2015, which is the most of any other baseball team.',
'The Bronx Zoo in New York is the largest city zoo in the United States with over 500 species and 4,000 animals.',
'In the 16th century, six Native American nations formed the Iroquois League in New York State, creating the oldest active democracy in the world. Many ideas were taken from it to form the government of the United States, such as federalism, elected representatives, caucuses, and impeachment.',
'The popular belief that Dutch settlers bought Manhattan Island from Natives in 1626 comes from a letter called the "Schaghenbrief" composed by Pieter Schager to leaders in Amsterdam. While the letter does mention that the island was purchased for 60 guilders (about US$24), it does not mention any involvement of the director of the colony, Peter Minuit.',
'In 1664, the English took the territory of New Amsterdam from the Dutch settlers living there. King Charles II named the territory New York after his brother the Duke of York and gave it to him as a gift.',
'New York was the 11th state in the United States and was a part of the original 13 colonies. New York City was the first capital of the nation from 1789 to 1790, and George Washington was inaugurated as the first U.S. president in the city's Federal Hall on Wall Street on April 30, 1789.',
'A third of the military engagements during the Revolutionary War took place in New York. During the war, 30,00 people left New York to escape the conflicts.',
'The New York Public Library has over 50 million books and other items and is the second largest library system in the nation after the Library of Congress. It is also the 3rd largest library in the world.',
'The first railroad in the United States ran 11 miles in New York, from Albany to Schenectady.',
'During WWII, President Franklin D. Roosevelt invited 982 refugees to stay in a holocaust refugee shelter called Safe Haven in Oswego, NY. The refugees consisted of survivors of concentration camps that had skills that would contribute to the refugee shelter. The president also had to promise that the refugees would return to their home country after the war, but many of the families gained clearance to stay in America.',
'Oneida, NY, was home to the Oneida Community from 1848 to 1880, a utopian society founded by John Humphreys. The 300 members of the community practiced abandonment of the self for the good of the whole, with men and women working side by side sharing all labor, property, and responsibility. Complex marriage was practiced, with each man being the

```
husband of every woman and vice versa, and reproduction was heavily monitored with certain men and women being chosen to have children together. The children stayed with their moms until they could walk, at which time they would be placed in the community nursery.',
'After WWII, the United Nations headquarters was established in New York City in 1952.',
'The New York Post, established by Alexander Hamilton in 1803, is the longest-running newspaper in the United States.',
'According to a report by CNBC, New York was the 3rd most expensive state to live in for the year 2014, after Hawaii and Connecticut. The average price of a home in New York that year was $1.3 million, the highest in the United States.',
'New York is the home of many inventions, including toilet paper and chewing gum. Rochester, NY, alone was the birthplace of marshmallows, Jell-O, Frenchs Mustard, baby shoes, gold teeth, mail chutes, and bloomers.',
'Famous New Yorkers include Billy Joel, Alicia Keys, Jennifer Lopez, Mariah Carey, Arthur Miller, Denzel Washington, Adam Sandler, Vera Wang, Kareem Abdul-Jabaar, Lou Gehrig, Eleanor Roosevelt, Theodore Roosevelt, Franklin D. Roosevelt, Lucille Ball, Tom Cruise, Washington Irving, Norman Rockwell, Michael Jordan, Herman Melville, Aaron Copland, George Gershwin, Maria Callas, Barbara Streisand, Walt Whitman, The Marx Brothers, and many more.',

];

//=========================================================================
=========================================================================
====
//Editing anything below this line might break your skill.
//=========================================================================
=========================================================================
====

const handlers = {
   'LaunchRequest': function () {
      this.emit('GetNewFactIntent');
   },
   'GetNewFactIntent': function () {
      const factArr = data;
      const factIndex = Math.floor(Math.random() * factArr.length);
      const randomFact = factArr[factIndex];
      const speechOutput = GET_FACT_MESSAGE + randomFact;

      this.attributes.lastSpeech = randomFact // by adding the last fact to the session attributes, we can use it to repeat it if asked by the user.
      this.response.cardRenderer(SKILL_NAME, randomFact);
      this.response.speak(speechOutput + " Do you want to hear another fact?").listen(" Do you want to hear another fact?"); // addig the  + " Do you want to hear another fact?" section allows
```

```
the skill to continue. the .listen also added
        this.emit(':responseReady');
    },
    'AMAZON.HelpIntent': function () {
        const speechOutput = HELP_MESSAGE;
        const reprompt = HELP_REPROMPT;

        this.response.speak(speechOutput).listen(reprompt);
        this.emit(':responseReady');
    },
    'AMAZON.RepeatIntent': function () {
        this.response.speak(this.attributes.lastSpeech + " Do you want to hear another
fact?").listen(" Do you want to hear another fact?"); // addig the  + " Do you want to hear
another fact?" section allows the skill to continue. the .listen also added
        this.emit(':responseReady');
    },
     'AMAZON.YesIntent': function () {
        this.emit('GetNewFactIntent');
    },
    'AMAZON.NoIntent': function () {
        this.response.speak(STOP_MESSAGE);
        this.emit(':responseReady');
    },
    'AMAZON.CancelIntent': function () {
        this.response.speak(STOP_MESSAGE);
        this.emit(':responseReady');
    },
    'AMAZON.StopIntent': function () {
        this.response.speak(STOP_MESSAGE);
        this.emit(':responseReady');
    },
};

exports.handler = function (event, context, callback) {
    const alexa = Alexa.handler(event, context, callback);
    alexa.APP_ID = APP_ID;
    alexa.registerHandlers(handlers);
    alexa.execute();
};
```

AVS Sample APP Code

.asoundrc

```
pcm.!default {
  type asym
   playback.pcm {
     type plug
     slave.pcm "hw:0,0"
   }
   capture.pcm {
     type plug
     slave.pcm "hw:1,0"
   }
}
```

initial_nala_startup.sh

```
#!/bin/bash

#Start companion service
echo "Starting companion Service"
cd /home/pi/Desktop/alexa-avs-sample-app/samples
cd companionService && npm start&



#Run the sample app
echo "Starting sample app."
cd /home/pi/Desktop/alexa-avs-sample-app/samples
cd javaclient && mvn exec:exec&

read -n1 -r -p "Press space to continue..." key

#Run the Wake Word Engine
cd /home/pi/Desktop/alexa-avs-sample-app/samples
cd wakeWordAgent/src && ./wakeWordAgent -e kitt_ai &
```

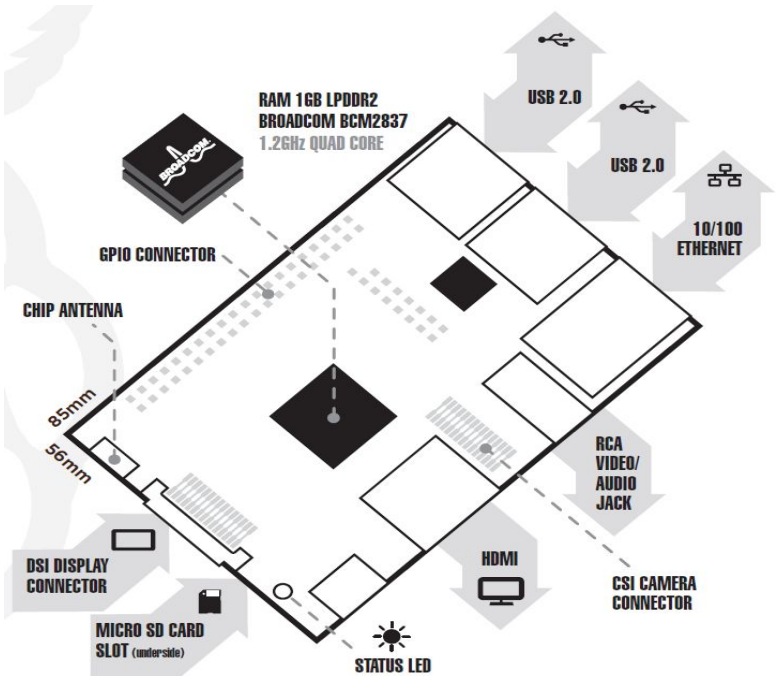
nala_startup.sh

```
#!/bin/bash

echo "Starting Companion Service"
cd /home/pi/Desktop/alexa-avs-sample-app/samples/
cd companionService && npm start&
echo "Starting Companion Service" > nala_startup.out

sleep 8

echo "Starting Javaclient"
cd /home/pi/Desktop/alexa-avs-sample-app/samples/
cd javaclient && mvn exec:exec&
echo "Starting Javaclient" >> nala_startup.out

sleep 28

#aplay /home/pi/Hello.wav

#sudo /home/pi/display16x32/bootLED.sh

#sleep 2

echo "Starting Wake Word Agent KITT-AI"
cd /home/pi/Desktop/alexa-avs-sample-app/samples/
cd wakeWordAgent/src && ./wakeWordAgent -e kitt_ai&
echo "Starting Wake Word Agent KITT-AI" >> nala_startup.out

aplay /home/pi/Hello.wav


sudo /home/pi/display16x32/bootLED.sh
```

Appendix B: Datasheets

Raspberry Pi 3

The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processor, 10x faster than the first generation Raspberry Pi. Additionally it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.



- Processor — Broadcom BCM2387 chipset.
  1.2GHz Quad-Core ARM Cortex-A53.
  802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
- GPU — Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infracture.
- Memory — 1GB LPDDR2

- Operating system — Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT.
- Dimensions — 85 x 56 x 17mm
- Power — Micro USB socket 5V1, 2.5A
- Ethernet — 10/100 BaseT Ethernet socket
- Video output — HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
- Audio output — Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
- GPIO connector — 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip. Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines.
- Camera connector — 15-pin MIPI Camera Serial Interface (CSI-2)
- Display connector — Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane.
- Memory card slot — Push/pull Micro SDIO

[data source: www.rs-components.com/raspberrypi]

Medium 16x32 RGB LED matrix panel

They have 512 bright RGB LEDs arranged in a 16x32 grid on the front. On the back there is a PCB with two IDC connectors (one input, one output: in theory you can chain these together) and 12 16-bit latches that allow you to drive the display with a 1:8 scan rate.

These displays are 'chainable' - connect one output to the next input - but our Arduino example code does not support this (yet). It requires a high speed processor and more RAM than the Arduino has. These panels require 12 digital pins (6 bit data, 6 bit control) and a good 5V supply, up to 2A per panel.

Keep in mind that these displays are designed to be driven by FPGAs or other high speed processors: they do not have built in PWM control of any kind. Instead, you're supposed to redraw the screen over and over to 'manually' PWM the whole thing. On a 16 MHz arduino, we managed to squeeze 12-bit color (4096 colors) with 20% CPU usage but this display would really shine if driven by any FPGA, CPLD, Propeller, XMOS or other high speed multi-core controller. The good news is that the display is pre-white balanced with nice uniformity so if you turn on all the LEDs its not a particularly tinted white.

- Dimensions: 192mm x 96mm x 12mm (7.6" x 3.8" x 0.5")
- Panel weight with IDC cable and power cable: 170 g
- 5V regulated power input, 2.5A max (all LEDs on)
- 5V data logic level input
- 2000 mcd LEDs on 6mm pitch
- 1/8 scan rate
- Indoor display, 150 degree visibility
- Displays are 'chainable' - connect one output to the next input - but our Arduino example code does not support this yet

[data source: https://www.adafruit.com/product/420]

Half-size breadboard

- 2.2" x 3.4" (5.5 cm x 8.5 cm)
- 9.7mm(0.38in) thick, including sticky foam on the bottom
- Weight: 38.9g(1.27oz)

[data source: https://www.adafruit.com/product/64]

TDA2822M Amplifier

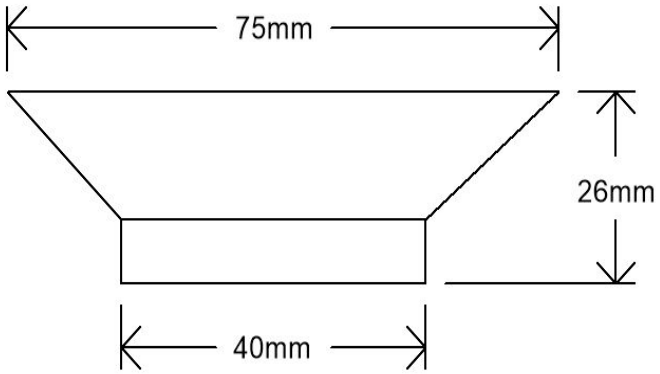
- Supply voltage: DC 1.8-12V
- Output Power: 0.1W-5W ;4-32 Ohms
- Dimensions: 52 x 31 x 23mm (L*W*H)
- Input Interface: 3.5mm Audio input
- Adjust form: volume adjustment

Using TDA2822M acts as amplifier chip, this chip is STMicroelectronics (ST) developed dual-channel monolithic power amplifier IC, usually in the pocket cassette player (WALKMAN), tape recorders and multimedia speaker in for the audio amplifier. Simple circuit, good sound quality and wide voltage range

[data source: https://www.amazon.com/gp/product/B076HKXLZT/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1]

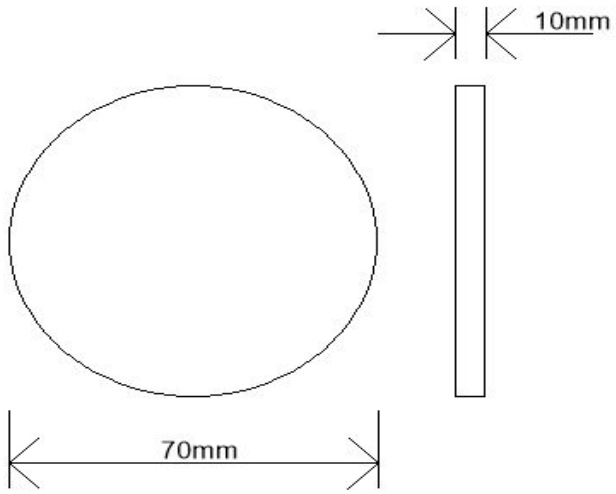
Speaker

- Impedance = 4 Ohms
- Power = 5 Watts



Microphone

- Strong Voice Reception Ability - Insert the USB plug before you use. Omni-directional microphone can pick up 10FT sound, 10 times higher than the traditional computer microphone.It is plug and play in normal.If there is no response after you plug the mic,please go to the microphone setting of your computer and choose using the usb microphone .
- Clear Audio - The high quality voice signal transmission function makes it can capture a slight signal, can clearly and vividly reproduce and expand your voice.
- Small and exquisite - Only 7 cm in diameter. Excellent round condenser microphone for the computer, pick up the sound from 360 degrees all-round , saving your space, can be placed on the desktop or pocket.
- Environmental Protection Materials - NASUM conference microphone uses the most advanced electrophoresis technology,adopts durable metal filter and high quality polyethylene materials.The metal protective shield will never fade.
- Applications - Accordiong to test, NASUM USB plug computer microphone is more stable than the microphone with 3.5mm plug .The USB computer microphone is perfect for video conferencing,office meetings, Skype, audio chat and other VoIP calls.
- Sample Rate: 32kHz
- Mic jack: USB

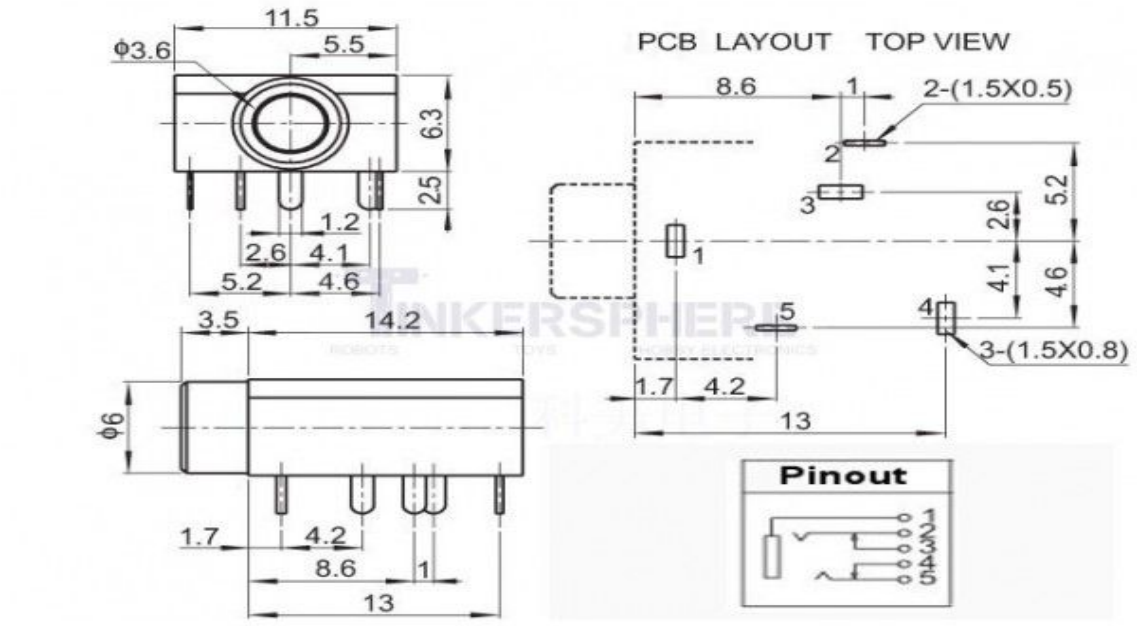


[data source:
https://www.amazon.com/Conference-Microphone-NASUM-Computers-Omnidirectional/dp/B076VV3M7K/ref=sr_1_1_sspa?ie=UTF8&qid=1526445097&sr=8-1-spons&keywords=nasum+microphone&psc=1]

Audio Jack

Use this 1/8" Panel Mount Female Stereo Audio Jack to add 1/8" input or output to your circuits. Unlike the 1/8" Female Stereo Audio Jack, this Headphone Jack is made for mounting on a printed circuit board for a permanent audio connection.
Specifications:

- Plug Size: 1/8" (3.5mm)
- Plug Type: Female
- 3 contacts (Left Audio, Right Audio & Ground)
- Clear top case for easy pin identification
- Plastic Body
- Tip with solder contacts screws into the Body



[data source: http://tinkersphere.com/audio/481-1-8-3-5mm-stereo-audio-jack-panel-mount.html]

Ground loop noise isolator

- Eliminating buzzing noise : Completely eliminating the buzzing noise, caused by ground loops which happens when the audio source and the speaker use the same power source in some car speakers / home stereo systems when using the Bluetooth receiver.
- Working principle : The working principle of this noise isolator is to achieve a clear speech/music by eliminating the current noise in some car speakers / home stereo systems.
- Compatible models : Works with any portable device that has 3.5mm audio jacks, for your Car Audio System/Home Stereo, when grounding issues persist. Also used with a Bluetooth Receiver/Bluetooth Hands-free Car Kit in your Car Audio System/Home Stereo.
- Compact and portable: Being so mini and light-weight (2.01*0.59*0.59 IN, 0.99 OZ), this little gadget does not take much space and can be easily taken away.

[data source:
https://www.amazon.com/gp/product/B019393MV2/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1]

LM2596S Buck Converter

- Input: DC 3 V to 40 V
- Output: DC 1.5 V to 35 V voltage is continuously adjustable, maximum output current is 3 A
- All use SANYO solid-state capacitors, 36 u thickening circuit boards, and high-Q inductance with LED indicator output Ground side is a direct 0 ohm pass through, aka common ground. Does not float or isolate
- Features :

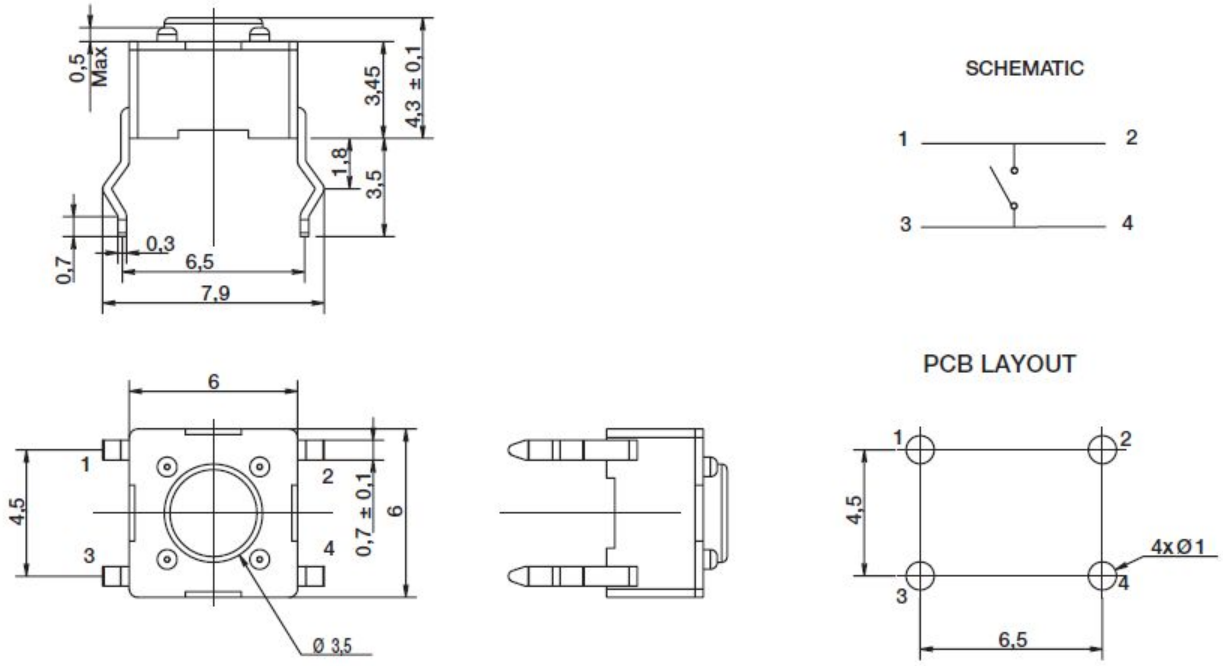
1. Physical size = 1.7" L x .830" H x .575" D (allowing .035" clearance for back of board - which is LIVE and must be insulated in conductive situations with poor clearance)
2. Mounting holes: Two (2) @ .108" (7/64) holes are staggered and offset .275" x .100 from the end and edge of the board. (FYI "end" means the long axis and "edge" means the short axis)
3. Trimmer is offset .975" from the left end of board, and .100" from the edge, with trimmer oriented to the top
4. Accepts via solder connection max 24 gauge wire
5. VDC I/O locations: (viewing board from front with trimmer UP!) B+ input, left upper hole. Reg. B+ out, right upper hole. B- (common!) left and right lower holes

- Steps: Connect to power source, adjust the swirl button on the blue potentiometer and monitor voltage with multimeter to get the needed voltage
- Warm tips: The input and output interface cannot be reversed. Keep it under 2.5 A and use the heat sink when working for long hours. To ensure stable output, input voltage must be 1.5 V higher than the output voltage

[data source: https://www.amazon.com/gp/product/B0748FKKWL/ref=oh_aui_detailpage_o05_s00?ie=UTF8&psc=1]

Power button

- Function: Momentary action
- Contact arrangement: SPST, N.O.
- Terminals: PC pins
- Actuation force: 130 grams, 160 grams, 200 grams, 260 grams
- Life expectancy: 100,000 operations.
- Contact rating: 50mA @12V DC
- Dielectric strength: 250 V AC min.
- Contact resistance: 100 mΩ max.
- initial. Insulation resistance: 10^{11} Ω min.
- Operating temperature: - 20˚C to 60˚C



[data source: https://www.ckswitches.com/media/1471/pts645.pdf]

12V/10A Power Supply

- Specifications:
  - Type: 12V 10A
  - Input: 100-240V AC 50/60Hz
  - Output: 12V 10A
  - Cable length: about 110CM
  - Output adaptor: 5.5mm x 2.1mm
  - OPower Cord is included:1.2M
- Features:
  - Simply & Professional appearance for power cabling
  - 5.5mm x 2.1mm Female power plug connector
  - Internal diameter of the plug: 2.1mm
  - External diameter of the plug: 5.5mm

[data source:
https://www.amazon.com/gp/product/B0111A92BK/ref=oh_aui_detailpage_o06_s01?ie=UTF8&psc=1]

5V/3A Power Supply

Just plug into any standard wall outlet and you'll have a solid 5V 3A power to the microUSB plug. This works great for powering your Raspberry Pi when using other peripherals that draw more current as it supplies 3 Amps of current! Most Raspberry Pi power supplies only supply 1 Amp which will cause power issues.

- Input Voltage: US 110 - 240 VAC 50/60Hz(wall outlet)
- Output Voltage: 5V
- Output Type: micro USB
- Current: 3A
- Cable Length: 3.9 feet
- FCC & CE Certified

[data source:
http://tinkersphere.com/raspberry-pi-accessories/2178-high-current-raspberry-pi-power-supply-5v-3a.html]

Appendix C:

Python Code:

```
import RPi.GPIO as GPIO
import time

delay = 0.000001

GPIO.setmode(GPIO.BCM)
red1_pin = 23
green1_pin = 13
blue1_pin = 5
red2_pin = 24
green2_pin = 21
blue2_pin = 19
clock_pin = 11
a_pin = 15
b_pin = 16
c_pin = 18
latch_pin = 7
oe_pin = 12

GPIO.setup(red1_pin, GPIO.OUT)
GPIO.setup(green1_pin, GPIO.OUT)
GPIO.setup(blue1_pin, GPIO.OUT)
GPIO.setup(red2_pin, GPIO.OUT)
GPIO.setup(green2_pin, GPIO.OUT)
GPIO.setup(blue2_pin, GPIO.OUT)
GPIO.setup(clock_pin, GPIO.OUT)
GPIO.setup(a_pin, GPIO.OUT)
GPIO.setup(b_pin, GPIO.OUT)
GPIO.setup(c_pin, GPIO.OUT)
GPIO.setup(latch_pin, GPIO.OUT)
GPIO.setup(oe_pin, GPIO.OUT)

screen = [[0 for x in xrange(32)] for x in xrange(16)]

def clock():
    GPIO.output(clock_pin, 1)
    GPIO.output(clock_pin, 0)

def latch():
    GPIO.output(latch_pin, 1)
    GPIO.output(latch_pin, 0)

def bits_from_int(x):
```

```
    a_bit = x & 1
    b_bit = x & 2
    c_bit = x & 4
    return (a_bit, b_bit, c_bit)

def set_row(row):
    #time.sleep(delay)
    a_bit, b_bit, c_bit = bits_from_int(row)
    GPIO.output(a_pin, a_bit)
    GPIO.output(b_pin, b_bit)
    GPIO.output(c_pin, c_bit)
    #time.sleep(delay)

def set_color_top(color):
    #time.sleep(delay)
    red, green, blue = bits_from_int(color)
    GPIO.output(red1_pin, red)
    GPIO.output(green1_pin, green)
    GPIO.output(blue1_pin, blue)
    #time.sleep(delay)

def set_color_bottom(color):
    #time.sleep(delay)
    red, green, blue = bits_from_int(color)
    GPIO.output(red2_pin, red)
    GPIO.output(green2_pin, green)
    GPIO.output(blue2_pin, blue)
    #time.sleep(delay)

def refresh():
    for row in range(8):
        GPIO.output(oe_pin, 1)
        set_color_top(0)
        set_row(row)
        #time.sleep(delay)
        for col in range(32):
            set_color_top(screen[row][col])
            set_color_bottom(screen[row+8][col])
            clock()
        #GPIO.output(oe_pin, 0)
        latch()
        GPIO.output(oe_pin, 0)
        time.sleep(delay)

def fill_rectangle(x1, y1, x2, y2, color):
    for x in range(x1, x2):
        for y in range(y1, y2):
            screen[y][x] = color
```

```
92.
93.
94. def set_pixel(x, y, color):
95.    screen[y][x] = color
96.
97. fill_rectangle(0, 0, 12, 12, 1)
98. fill_rectangle(20, 4, 30, 15, 2)
99. fill_rectangle(15, 0, 19, 7, 7)
100.
101.   while True:
102.      refresh()
```

C/C++ Code:

Running some demos

$ make

$ sudo ./demo

usage: ./demo <options> -D <demo-nr> [optional parameter]

Options:

-D <demo-nr> : Always needs to be set

-t <seconds> : Run for these number of seconds, then exit.

--led-gpio-mapping=<name> : Name of GPIO mapping used. Default "regular"

--led-rows=<rows> : Panel rows. Typically 8, 16, 32 or 64. (Default: 32).

--led-cols=<cols> : Panel columns. Typically 32 or 64. (Default: 32).

--led-chain=<chained> : Number of daisy-chained panels. (Default: 1).

--led-parallel=<parallel> : Parallel chains. range=1..3 (Default: 1).

--led-multiplexing=<0..6> : Mux type: 0=direct; 1=Stripe; 2=Checkered; 3=Spiral; 4=ZStripe; 5=ZnMirrorZStripe; 6=coreman (Default: 0)

--led-pixel-mapper : Semicolon-separated list of pixel-mappers to arrange pixels.
Optional params after a colon e.g. "U-mapper;Rotate:90"
Available: "Rotate", "U-mapper". Default: ""

--led-pwm-bits=<1..11> : PWM bits (Default: 11).

--led-brightness=<percent>: Brightness in percent (Default: 100).

--led-scan-mode=<0..1> : 0 = progressive; 1 = interlaced (Default: 0).

--led-row-addr-type=<0..2>: 0 = default; 1 = AB-addressed panels; 2 = direct row select(Default: 0).

--led-show-refresh : Show refresh rate.

--led-inverse : Switch if your matrix has inverse colors on.

--led-rgb-sequence : Switch if your matrix has led colors swapped (Default: "RGB")

--led-pwm-lsb-nanoseconds : PWM Nanoseconds for LSB (Default: 130)

--led-no-hardware-pulse : Don't use hardware pin-pulse generation.

--led-slowdown-gpio=<0..2>: Slowdown GPIO. Needed for faster Pis/slower panels (Default: 1).

--led-daemon : Make the process run in the background as daemon.

--led-no-drop-privs : Don't drop privileges from 'root' after initializing the hardware.

Demos, choosen with -D

0 - some rotating square

1 - forward scrolling an image (-m <scroll-ms>)

2 - backward scrolling an image (-m <scroll-ms>)
3 - test image: a square
4 - Pulsing color
5 - Grayscale Block
6 - Abelian sandpile model (-m <time-step-ms>)
7 - Conway's game of life (-m <time-step-ms>)
8 - Langton's ant (-m <time-step-ms>)
9 - Volume bars (-m <time-step-ms>)
10 - Evolution of color (-m <time-step-ms>)
11 - Brightness pulse generator

Example:

./demo -t 10 -D 1 runtext.ppm

Scrolls the runtext for 10 seconds

REFERENCES

[1] H. Robinson, B. Macdonald, N. Kerse, and E. Broadbent, “The Psychosocial Effects of a Companion Robot: A Randomized Controlled Trial,” *Journal of the American Medical Directors Association*, vol. 14, no. 9, pp. 661–667, 2013.

[2] AndrewH7, “Simple Raspberry Pi Shutdown Button,” *instructables*, 24-May-2015. [Online]. Available: http://www.instructables.com/id/Simple-Raspberry-Pi-Shutdown-Button/.

[3] Benchoff Brian, “Introduction to the Raspberry Pi 3,” *Hackaday*, 3-May-2018. [Online]. Available: https://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/

[4] Hzeller, “rpi-rgb-led-matrix,” *github*, 29-August-2017. [Online]. Available: https://github.com/hzeller/rpi-rgb-led-matrix/blob/master/wiring.md

[5] Tyler, “How to add a power button to your raspberry pi,” *howchoo*. [Online]. Available: https://howchoo.com/g/mwnlytk3zmm/how-to-add-a-power-button-to-your-raspberry-pi

[6] Eirk, “alexa/alexa-avs-sample-app,” *GitHub*. [Online]. Available: https://github.com/alexa/alexa-avs-sample-app/wiki/Raspberry-Pi

[7] “Alexa Skills,” *Amazon*. [Online]. Available: https://www.amazon.com/alexa-skills/b/ref=topnav_storetab_a2s?ie=UTF8&node=13727921011