INFO 4320 Final Report: NFC-Enabled Jukebox

Yaoyao Ma
Cornell University
Ithaca, United States
ym284@cornell.edu

Yingjie Zhao
Cornell University
Ithaca, United States
yz483@cornell.edu

Javier Vega
Cornell University
Ithaca, United States
jdv67@cornell.edu

ABSTRACT
UPDATED—15 May 2022. This paper describes the NFC-Enabled Jukebox project done for INFO 4320. The goal for this project is to develop an automated music player.. Users can pick a song to play by interacting with a LCD screen. A motor-powered wheel with the NFC cards rotates in sequence and stops whenever a song is selected. The associated NFC card is then pushed up. The user then scans it through the top of the machine and the song chosen will immediately play from the speakers on the side. Volume for the song is adjustable as well.

Author Keywords
Rotators; Motors; Interfaces; Music; Scanner; Arduino; Stepper; NFC; Servo; User interface.

ACM Classification Keywords
•Hardware~Electronic design automation~Physical design (EDA)~Placement

INTRODUCTION
The machine will use NFC cards to store song names in a rotating wheel with a box that also serves as a container. Users can request specific songs by interacting with the digital screen on top of the machine. When instructed by the user, the wheel rotates, presenting a new song the user can select. Once a song is selected by the user, its NFC card would be pushed up for the user to grab and scan to play. Music instantly comes out from the speaker once the card is scanned. The user then puts the card back and can enjoy their music.

The goal of this project is to provide music lovers with physical interaction through a nostalgic jukebox machine that implements the latest technology and yet calls back to the times of vinyl, cassette and CDs, echoing the wave of new-retro culture in contemporary society while being technologically up to date.

Our machine is with a transparent acrylic top cover for people to see through the inside that holds the music cards. The NFC scanner on the top of this machine communicates with the local SD card in the audio shield that stores 10 song files. Rotary navigation wheel, buttons and scanners follow a minimalist design approach so that the functionalities around scanning NFC cards and operating the machine would be simple for users to understand and creates an effortless workflow.

Description
Our design can be segmented into three distinct parts: the physical rotary storage system, interactive input system, and the music decoder / signal amplifier. The interactive input system allows users to interact with an LCD display plus basic control buttons that triggers rotation of the card navigation wheel as they select through the song. The physical storage system has two stepper motors, one controls the spinning wheel containing NFC cards for scan. The other one pushes up (pops out) the correct card when selected, by using an eccentric sheave with an eccentric rod fitted. When the card has been pushed up, a distance sensor underneath that card will send out a signal to the microcontroller in order to tell the state of that card (popped out/pushed in), then the microcontroller will send the corresponding signal to the motor driver to control turning. The music decoder / signal amplifier will be done on an arduino music shield, it reads the SD card and maintains a list of file names, when the top NFC scanner gets the radio wave and retrieved data from a card, the microcontroller will lookup the song name in local SD card and send commands to the music shield which automatically plays the song and do signal amplify for us via two 8 ohms speakers connected to the music shield.

RELATED WORK
A carousel of cassette [1] inspired us to come up with the jukebox idea, especially the logic behind the rotator mechanism. Instead of cassettes, we want to use NFC [8] cards which connect to the barcode technology that is seen commonly in our current daily life. We also like the idea of implementing a volume button, just like Juuke [2]. We intend to use an LCD display with RGB backlight [3][7] to display song information and play status, and serve as an interface for user interaction and commands with control buttons. We also drew our inspiration from a smart rack inventory system [4] where RFID and Arduino are used to build the connection. Our wires would be organized neatly so that the risk of the machine being interrupted in the middle of running would be lower. Our music would be played after getting decoded by a simple yet effective music make shield [5]. The volume control feature should be realized through an intuitive scroll wheel rotary encoder that perfectly reminisces the original clicking scroll wheel interface on the first iPods [6]. Two LED bars [9] are placed to the sides of the volume control wheel to display the change through decreasing/increasing light blocks, including color codes that indicate safe hearing levels in green and the opposite in red on top.

INITIAL PROTOTYPE

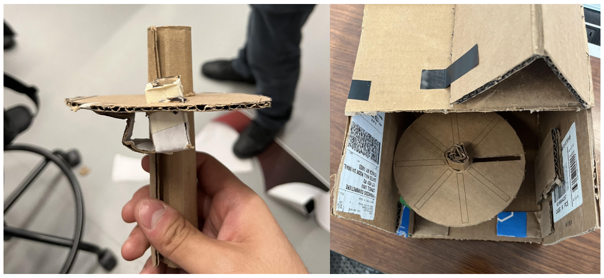

Figure 1&2. Motor / wheel storage system design.

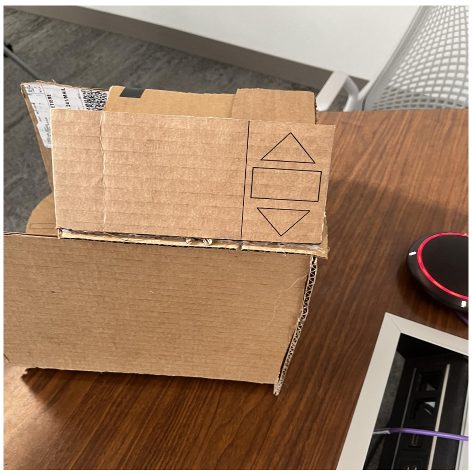

Figure 3. User Interaction Interface.

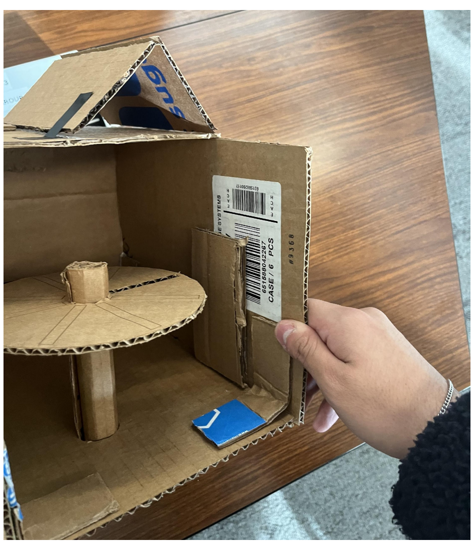

Figure 4. Overall Structure.

For the initial prototype, we were unsure on how to push the card structure up. Figure 1 shows the motor / wheel storage system design. There will be a central axle, and a wheel with slots with a card holder in it. These card holders will hold the NFC cards. We will likely use a stepper motor to ensure precise movements, since the card holders need to be in exact positions to push the card up through the top of the machine. We decided that the cards would be pushed by one piston that pushed the entire holder up. The NFC reader is shown in the second picture as the cardboard piece on the right perpendicular to the card holder. Figure 3 shows the design for the user interaction interface. It will have buttons for up and down and select. The up and down buttons are to move between songs, and when the user finds a song they like, they will select it. Then, the storage system will find the right card and retrieve it for them. The physical interface will use an LCD screen. Figure 3 also shows the NFC music player in the back. It's a simple flat surface which the user will tap the card on.

Figure 5. Modified design for pushing the card up through the slot.

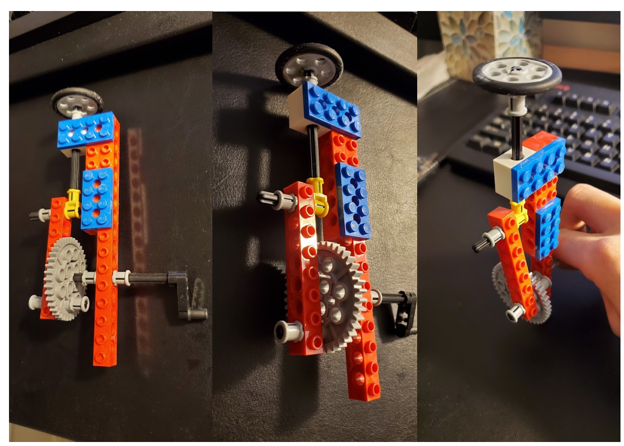

Figure 6&7&8. Details of Lego piston mechanism structure.

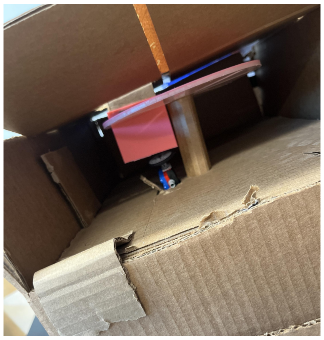

Figure 9. Lego piston pushes up the card holder.

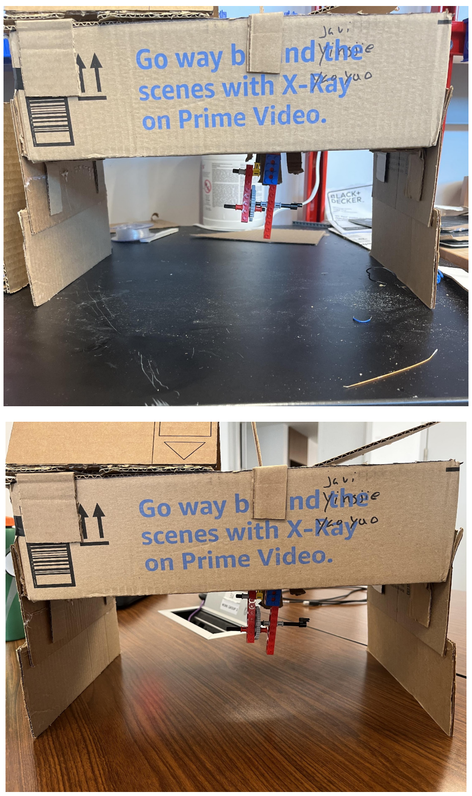

Figure 10&11. Front view with Lego piston structure installed inside.

Figure 12. Top view of the whole machine.

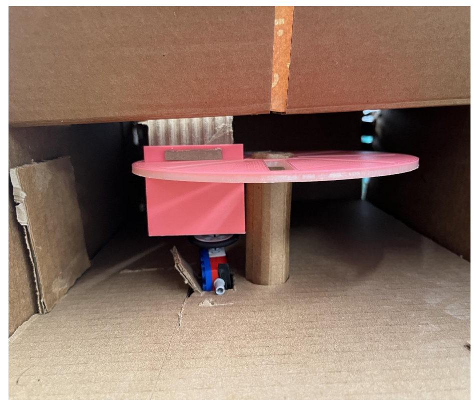

Figure 13. Side view of the turntable and card holder with Lego piston push-up mechanism.

Figure 5 shows a full image of our final prototype. It's a simple box with the physical interface and scanner on the outside, but the storage system on the inside. There will be one opening in the box to let the selected card through.

Figure 9 shows our modified design for pushing the card up through the slot. The wheel storage will have card holders, and a piston, modeled by the Lego structure shown in figure 6&7&8. It will push the card up for the user to grab and use. Another view of the Lego structure is available in figures 10&11, 12, and 13.

The physical system is similar to the initial prototype, and so is the scanner.

FUNCTIONAL UNITS

Our final design will still fall into three major components.

Storage System

The most challenging component is still the storage system. To maximize the stepper motor's utility, the wheel storage will have to store several cards divisible by 200, the number of steps in a stepper motor. Therefore we decided on 10 cards. Additionally, scanning the cards will prove to be challenging, but we are hoping to use cards of at least 6 mm diameter so the scanner can be perpendicular to the card holders for motion simplicity. We are happy with the piston/upward pushing structure driven by a servo. We are considering putting springs in the card holder, which are released when the right card is identified, pushing the card up to the user. This could prove to be challenging, and so we may stick with the existing design.

Physical Interface

The physical interface should be straightforward, but some challenges exist. Our original plan for the interface was to have the user select their song first by using the left/right buttons, and the storage system would scan every card until the desired one was found. When the user finds a song they like, they press the up/down button, and the piston pushes that card up. We are leaning toward this because it improves the user experience. It would not be more difficult than our original design, just change how we set up the code and buttons.

Scanner

The scanner presents one major challenge and depends on how much time we have and how complex we want our system to be. Since we plan to play music from the SD card, the nfc cards will simply contain the song name and the artist name. So the scanner will read this info from the tag and parsing it to a filename, then pass the filename to the VS1053 chip in order to tell which mp3 file to play. Extending from this, we can either connect through an audio jack. Time permitting, we will do whatever results in the best user experience.

DEMO VIDEO LINK

https://youtu.be/2962P-2svLc

note: This report does not include explicit mention of our second prototype, the one due April 18 since a report was not required for that submission. For more on that, please refer to our submission from that day.

The overall takeaways and areas for improvement and further iteration we identified were:

- Finalize the "pusher" functionality; perhaps there's a better option than a servo.
- Switch the LCD screen for more space to include more info for the user.
- Play the music! connect to the song library, and get the scanner up and running.
- Switch microcontroller; we are out of pins and will need more for the final prototype.
- Add functionality to make sure the user puts the card back; probably use a line reader.
- Generally, debug and optimize the design.

FINAL DESIGN

Our final design is broken up into the mechanical part of the design, and then the electrical / code design part.

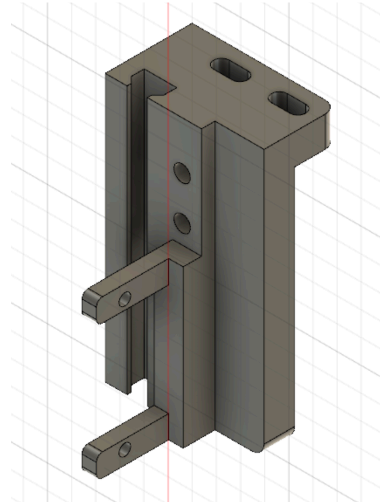

Figure 14: Motor bracket downloaded from the internet but modified by us.

Figure 14 shows the motor bracket we used to hold the pusher for the pusher mechanism. We edited the design to fit with our servo motor. Here is the source: https://www.thingiverse.com/thing:3170748/files.

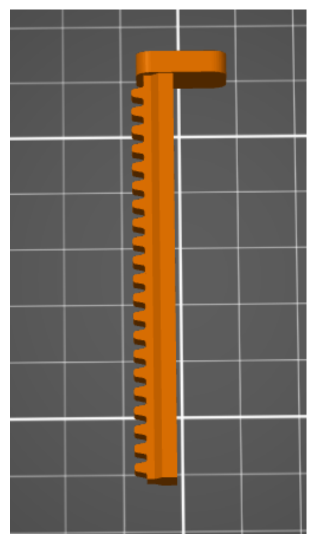

Figure 15: 75mm pusher

This is the accompanying 75 mm pusher from the download. We did not have to edit it.

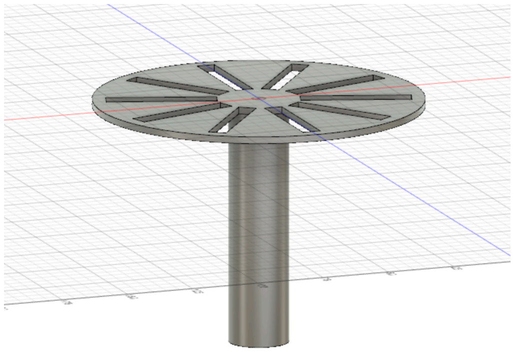

Figure 16: Final column/dial design

Figure 16 shows the final column and dial design. Even after iterating on this design, it was difficult to find a balance between a large enough dial, accuracy from the stepper motor, and putting too much weight on the motor. We will discuss this further in the limitations section.

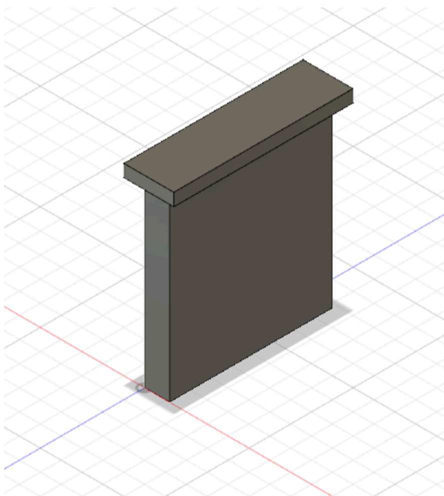

Figure 17: Final card structure

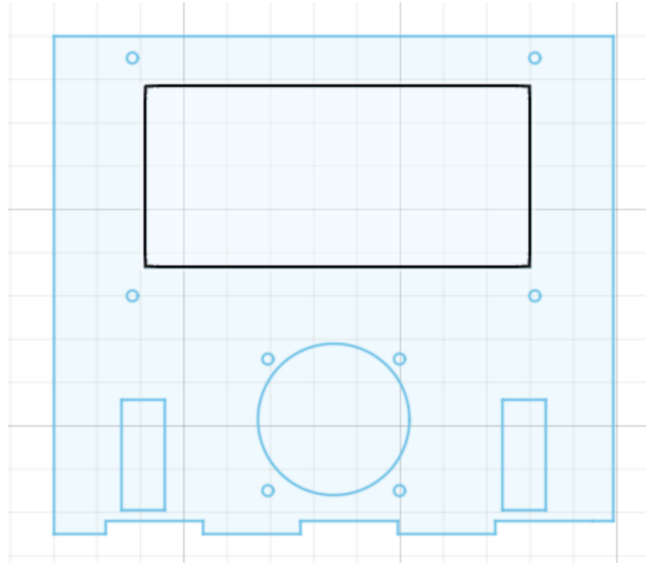

Figure 18: Final laser cut design for the physical interface

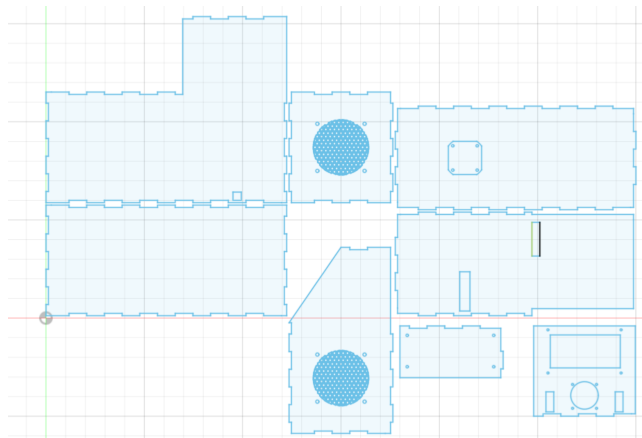

Figure 19: Final box layout

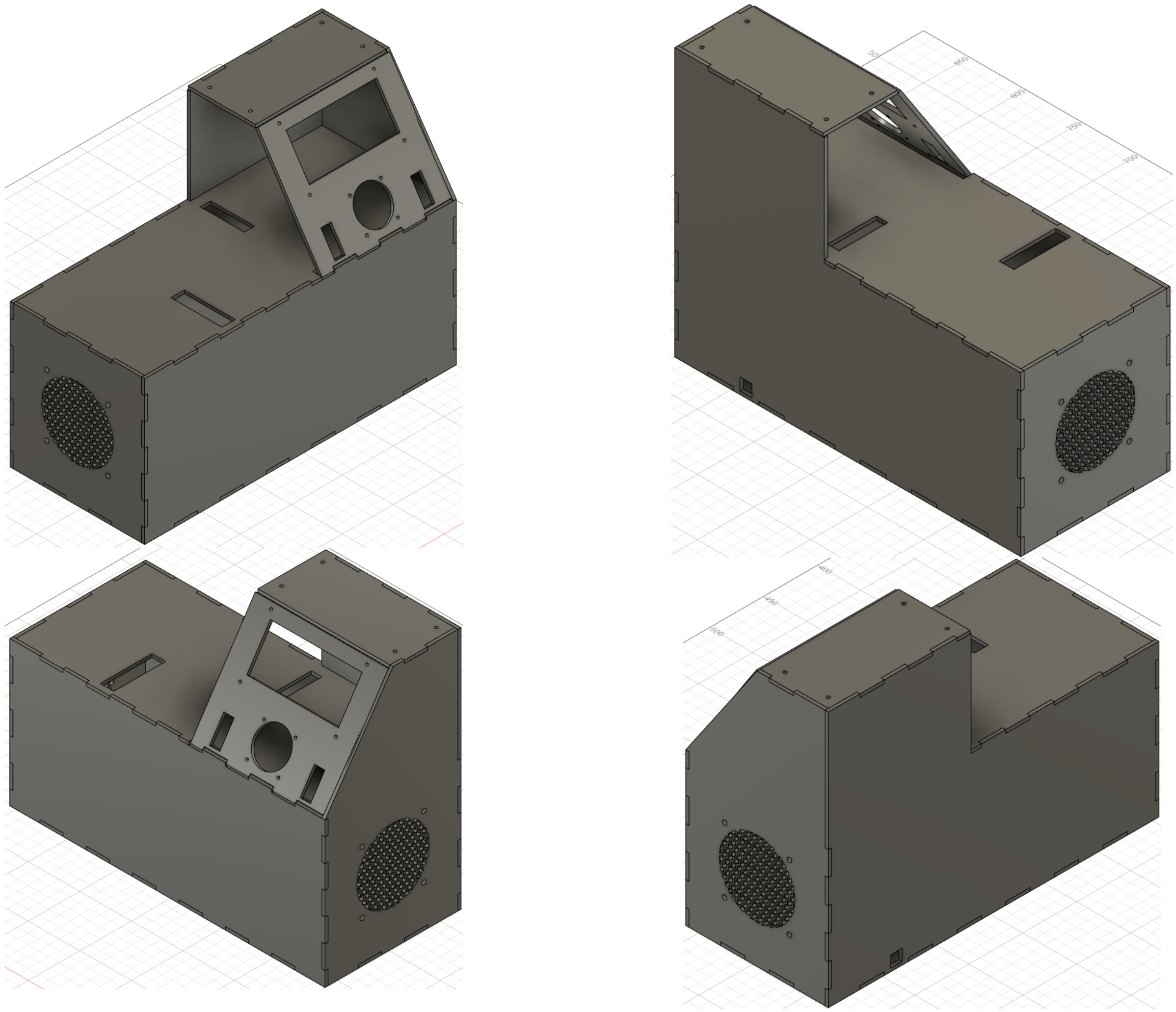

Figure 20: Final box model

This model does not take into account the interior of the box, and the box will be acrylic.

ELECTRICAL PARTS SELECTING

Microcontroller:

Initially we chose the SparkFun ReadBoard Artemis ATP as our main controller, since it interrupts on all pins and 3.3V logic, which is a friendly voltage to all sensors. However, our team is facing one critical issue which is that the NFC reader (PN532) library does not support Artemis CPU, the NFC reader library only supports Atmel and ESP32 CPU. Therefore, we switched to Arduino MEGA 2560, this board provides sufficient pins and Atmel CPU onboard with 5V logic, so there is no issue with the NFC reader. But a new problem is our ANO rotary encoder and navigation buttons need interrupts to trigger. However, according to the manual of Arduino MEGA 2560, only 6 pins support interrupts, and two of them reserved for I2C communication, here we need 5+2 = 7 interrupt pins. Without interrupts, our buttons and rotary encoders will have ghost effects and are not accurate.

Finally, we found the Arduino Due is the right microcontroller, it's all pin interrupt, 3.3V logic also with Atmel CPU onboard. Although it's based on ARM Cortex-M3 CPU, which is a RISC architecture, there is no problem with sensor libraries.

Two NFC board with the same I2C address:

In our design, there are two NFC boards, one sitting near the wheel, used for scanning the card that is in the wheel; another one is on the top of the box, used for reading the card and playing the music. Since both NFC boards are using I2C communication with the same unchangeable address. To solve this problem, we use a [Digital 1-to-8 Multiplexer] to control them. Every time when we need to talk to one of the NFC boards, tell the multiplexer to select only one of them.

IMPLEMENTATION

Box Logic:

1) User interacts with interface, first NFC scanner scans card, reading song name for user through the display
2) User turns the dial using the left and right wheel buttons. Stepper turns 20 steps either way.
3) The first NFC scanner reads again, displaying a new song.
4) User likes the song, presses the up button to select the new song.
5) Microcontroller sends the song name to the second NFC reader. The second NFC reader waits for the user to scan the card to play.
6) Once the second reader scans a signal, the song starts playing. Users can control volume, play/pause with the center wheel.
7) When they are done with this song, in order to select a new song, they must return the old card. The box is waiting for the user to push the pusher back down and read the scan for the correct card. Until the scanner reads the card is back in place, the user cannot turn the dial.
8) Once the card is back in place, the user can select their next song, and repeat this logic.

Storage System

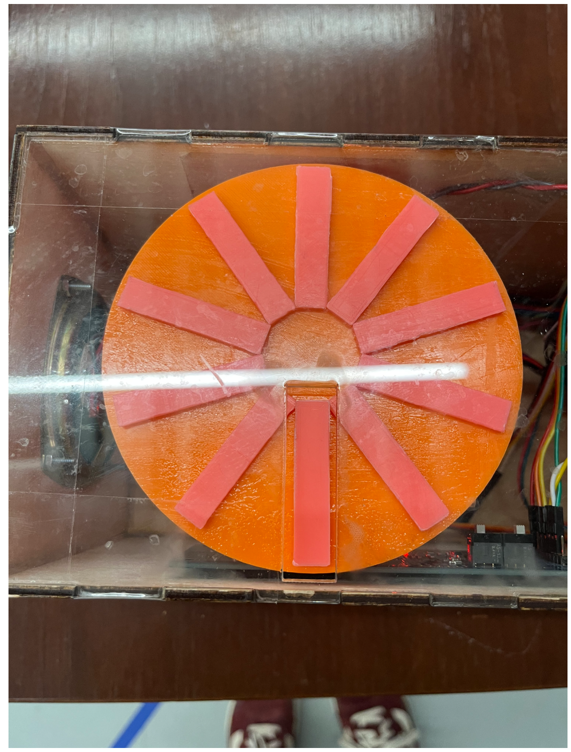

Figure 21: Final dial implementation with cards

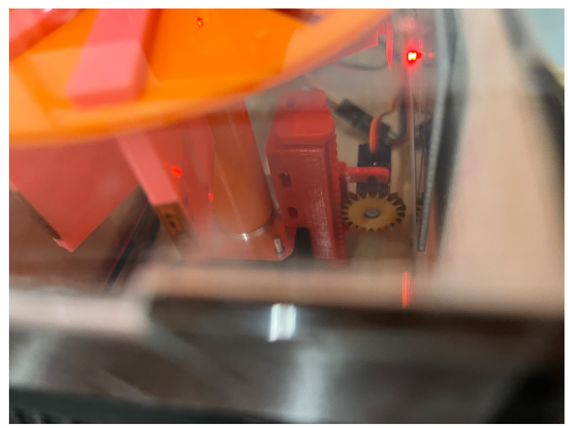

Figure 22: Final pusher implementation in the center, with the first scanner in view to the right and stepper at the bottom

The final storage system looked pretty similar to the final prototype design we had in mind. The dial has 10 slots for 10 cars holding the NFC chips. The dial is controlled by a stepper, which makes 20 steps whenever the user presses the turn button. We decided to use a linear actuator for the "pusher" mechanism, deviating from our original piston idea. We did this because the linear actuator would be controlled by a servo motor, which allows for a slower push upward, which is good for the user experience, as it looks measured and purposeful and prevents accidentally breaking the motor by force if something gets jammed.

When the user finds a song they like, the pusher gently pushes upward and provides them with the correct card.

When they are finished with the card, they cannot switch to the next card until the current card is returned.

Physical Interface

Figure 23: Final physical interface implementation

Our final implementation was almost identical to our final design. We were hoping to get a larger screen to include more information and we did exactly that. We also added volume measures to indicate to the user the volume level. The screen displays the artist and song when it scans a song, and when a song is loaded, it says if it is paused or playing, and the timestamp for the song.

Scanner

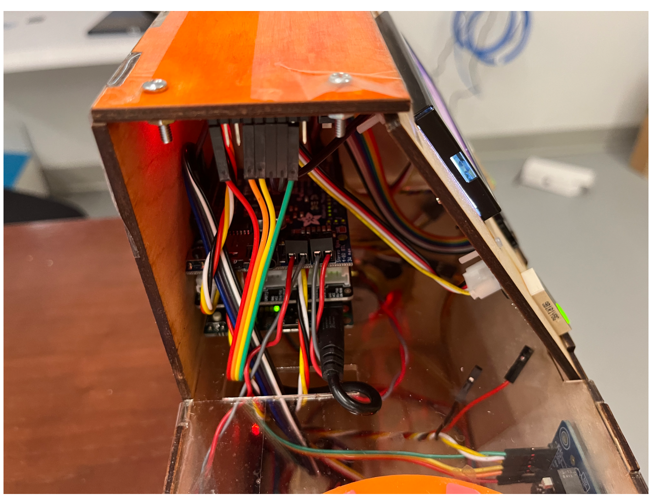

Figure 24: Scanner at the top, beneath the top plywood, and the speaker visible in the bottom of the background

The final implementation for the scanner was almost like we designed. The scanner would be underneath the top plywood for usability and aesthetic. After receiving the song, two 8 ohms resistors play the music with variable volume depending on user input.

On the software/program part, there are two state machines to track the current state. One for music (playing/paused/stopped), and the other for the piston (ejected/inserted). When the user wants to browse songs from cards, the music must be paused or stopped and the current card must be inserted into the wheel.

We implemented an algorithm in order to track the elapsed time when playing. The basic logic is when the NFC reader get the song information and passed it into the music shield to play, record the current time by using ***millis()***, everytime iterate the ***loop()*** function, calculate the time difference between the current one and the recorded one, use modulation to get how many seconds and use division to get how many minutes. When there is a pause command shows up, save the current time difference to a variable, then wait for a continue command. Once received the continue command, keep calculating the time difference between the resume time and the current time, then add the previous saved one.

LIMITATIONS AND FUTURE WORK

In the final testing stage, we realized that because of the weight of the NFC cards and an error interval encountered by the spinning angle in each cycle triggered by stepper motor, the wheel would always rotate more than expected, causing the sanner to miss scanning points and cards to get stuck when there is a pop-up attempt. We fixed this by manually re-calculating and re-implementing the algorithm based on the actual stepper motor spinning angle observed to be needed.

A constraint in our current scope is that the NFC scanner is not triggered by interrupt, meaning that once the card has been pushed up and scanned to play a song, there is no way of going back to choose another song instead, unless you let the song play and pause it before pushing the card back down and going into another selection cycle. This is the current limitation because the second NFC reader is configured to be waiting for a tag to advance the whole program code again. A future improvement that can be made is to add an auto-advance feature by rewriting some parts of the algorithm so that a smoother experience could be provided to users when they want to switch between various songs they like.

Eventually, we would like to upgrade the jukebox to hold song cards that are not only limited to one song each. With NFC cards of higher storage capacity, it would be ideal to store an entire album within one card, so that users can easily navigate between different songs from one record just like how they would with a CD or a cassette.

REFERENCES

1. Techmoan. Fascinating Vintage 20 Cassette Carousel from 1972 : Panasonic RS-296US. August 14, 2016 from https://www.youtube.com/watch?v=RJo13FP4UpI&t=96s

2. ARDUINO Team. Juuke is an Arduino-powered RFID music player for the elderly. August 1, 2020 from https://blog.arduino.cc/2020/08/01/juuke-is-an-arduino-powered-rfid-music-player-for-the-elderly/
3. SparkFun. SparkFun 20x4 SerLCD - RGB Backlight (Qwiic) from https://www.sparkfun.com/products/16398
4. LearnWithSHIE. Smart Rack Inventory System with RFID and Arduino. Jun 26, 2020 from https://www.youtube.com/watch?v=2GclXDMDZng
5. lady ada. Adafruit Music Maker Shield. Adafruit from https://learn.adafruit.com/adafruit-music-maker-shield-vs1053-mp3-wav-wave-ogg-vorbis-player
6. Kattni Rembor. ANO Directional Navigation and Scroll Wheel Rotary Encoder and Breakout. Adafruit from https://learn.adafruit.com/ano-rotary-encoder
7. QCPETE. AVR-Based Serial Enabled LCDs Hookup Guide. SparkFun from https://learn.sparkfun.com/tutorials/avr-based-serial-enabled-lcds-hookup-guide?_ga=2.219555993.1826968717.1652663455-1783388412.1651000137
8. lady ada. Adafruit PN532 RFID/NFC Breakout and Shield. Adafruit from https://learn.adafruit.com/adafruit-pn532-rfid-nfc
9. Seeed Studio. Grove-LED Bar from https://wiki.seeedstudio.com/Grove-LED_Bar