Spring 2022

Cardbot

Jakob Reiter
jmr288@uakron.edu

Grant Crump
gwc13@uakron.edu

Drake Drivere
dad173@uakron.edu

Adam Miner
alm134@uakron.edu

Senior Project Final Report

Design Project: Card Shuffling and Dealing Robot

Grant Crump

Drake Drivere

Adam Miner

Jakob Reiter

Dr. Sastry

Date Submitted
04/25/2022

Table of Contents (JR)

List of Figures

(GC)

List of Tables

(GC, DD)

Abstract

Many people enjoy playing cards but playing cards in the times of COVID-19 is difficult. There are many designs for dealing and shuffling robots, but few include sanitization and autonomous operation. This design is for a BlackJack card dealing and shuffling robot that will be autonomous and will sanitize the playing cards using UV-C LEDs. This will assist in keeping the user safe while playing a game of BlackJack. (DD)

Upon project completion, many physical or mechanical problems were encountered. Although each subsystem worked electrically and with code implementation, the system as a whole simply encountered too many mechanical problems and led to a robot which did not meet the outright specifications described herein. As a learning experience, this project was a massive success. (AM)

1. Problem Statement

1.1 Need

Playing cards is a popular activity. While some prefer to play with family and friends, others prefer to play in casinos. In both situations, cards must be shuffled, dealt, and collected after each round of play. These tasks are time consuming, inefficient, and introduce error in dealing/shuffling as well as the spread of germs via the handling of cards. For casinos specifically, humans handling cards introduces several opportunities for loss of productivity from breaks and shift changes. “In most casinos, the breaks for dealers are extremely generous, with most working for an hour and a half before being allowed up to half an hour break” [1]. This, as well as the associated swapping of dealers leads to time and money lost. Also, casinos could have a problem with the implementation of legally required policy with respect to the spread of germs, and various other types of human error. (GC, DD, AM, JR)

1.2 Objective

An automated card dealing robot will be designed and implemented. It will be able to shuffle and deal the cards for each round of play. It will also be able to accept verbal commands (such as “hit” or “stay” in blackjack). It will have several compartments for extra decks. It will have a UV sterilization chamber, within which all cards that are touched by humans will be sanitized. It will also be able to collect the cards without requiring any form of human-to-human contact, whether it be direct or indirect. (GC, DD, AM, JR)

1.3. Background:

The card shuffling and dealing robot will be made up of multiple systems. These systems will include a voice recognition system, a UV sterilization system, a shuffling system, and a dealing system. Each system will be explained in more detail along with background, examples, and implementation below. The purpose of these systems is to remove the need of a human dealer that is needed in the game Blackjack and will eliminate a large amount of human contact during said Blackjack game. This is especially relevant to the future of playing card games with the outbreak of Covid-19, thus making activities like this safe is very important. (JR)

One of the components to the Card shuffling and dealing robot is the ability to interpret voice commands from the user. These would include simple commands such as “Hit” and “Stand” in the game of Blackjack, for instance. The purpose of this voice recognition system is to simulate what it is like to be a part of a real game at a casino. There are multiple ways this technology is used and has been used. Most notably and currently, it is used in all kinds of smart phones in the form of Google Assistant in Android phones and Siri in Apple iPhones. Another example was found in a book called “Statistical Methods for Speech Recognition”. This book talks about a recording from a chess match in the 1960s in a project by Raj Reddy. “Recognition of spoken chess moves could be based on a small vocabulary, a restricted syntax, and a relatively complete word knowledge: The system knew which chess moves were legal, so that, for instance, a recognition hypothesis corresponding to a move of a piece to a square occupied by another piece of the same color could immediately be rejected.” [2]. The system that is planned to be used in the card shuffling and deal robot would use similar criteria that the chess system referred to above. There are certain words that can be used with specific games and all other words would be rejected. The voice commands that are planned are not complicated sentences, they are one-word commands that are simple and easy to understand. The system will have colored LEDs on it to signify whether a voice command was processed and understood or not; green for understood and red for not understood. (JR)

There are many ways to use voice recognition with electronics, as seen by the academic journal article, "Voice recognition system for controlling electrical appliances in smart hospital room" by Eva Inaiyah Agustin, Riky Tri Yunardi, and Akbar Firdaus [3]. A specific instance of this implementation uses voice recognition in a room where a smart hospital bed can move in multiple positions such as up or down and a lamp can be turned on or off by command of the user. This bed system can only recognize up to 7 voice commands that work at the same time. The voice command goes into a microphone, through a voice recognition module, Arduino Uno microcontroller and a driver relay circuit. If there are more than seven voice commands, the loading process is done using a different method. The card shuffling robot proposed herein will take in a one syllable word such as “hit” or “stay” from a maximum of 4 players. The robot will take the command into a microphone and run through the voice recognition system. Then, the robot will process this command and rotate to whichever player's turn it is, and deal or not deal based on the received command. Through this article's project, one syllable commands had an 85% or greater chance of implementing the command through using an Arduino Uno microcontroller. Within the scope of the proposed project, a PIC board will supplant the microcontroller. (GC)

The hardware and software associated with voice recognition will need to be designed and implemented for the automated card dealing robot. In the patent by Mark LaBosco [4], a system is described in depth for the reception of acoustic audio signals, the cancellation of noise during the reception process, the processing of the acoustic audio signals into digital signals, and the output of those digital signals. The system proposed herein will utilize the concepts described within that patent. A physical microphone will be used as the input transducer to the system. After the input will be a noise cancellation system, which will detect any reference noise (room conversation, internal noise, EMI, etc.). It will cancel these background signals such that only the spoken command will remain. After the noise cancellation system, a circuit will be implemented that turns the analog voice signal into a digital signal. After the analog signal is digitized, the digital signal will then be passed into a voice recognition system. This system will be set to receive only a set number of voice commands (hit, stay, etc) in an effort to reduce complexity of the system. After the command is recognized, the voice recognition system will then send one of any number of control signals to the control circuit of the automated card dealing robot, which will then perform the requested command. (AM)

The use of LEDs is an efficient way of delivering UV-C radiation to a localized area. As described in the application for the sanitization of toilet bowls, "LEDs require no warm-up time, less energy to operate, and exhibit a very long operational lifespan up to 100,000 hours" [5]. The approach for the sterilization chamber of the automated card dealing robot will therefore be implemented with a UV-C-LED circuit. The efficiency of UV-C-LED irradiation on a localized surface within a toilet bowl was obtained in the experiment cited above. The results were such that an 8-LED configuration was the most effective. Since the application cited was done in a very high contamination environment, an 8-LED configuration within the sterilization chamber of the automated card dealing robot should be sufficient. In the experimental results of Lai and Nunayon [5], they discovered that the irradiance of UV-C-LEDs was highest within 1 cm away from the source, so that distance should be sufficient for the project proposed herein (in this case, the source is the deck of cards). (AM)

According to the article by Dana Mackenzie [6], the COVID-19 virus is vulnerable to UV-C radiation. This is due to the fact that the UV-C radiation emitted by Earth's sun is actually absorbed in Earth's atmosphere, according to Mackenzie. This makes the COVID-19 causing virus ill equipped to combat an assault via UV-C radiation, due to a lack of exposure to that specific type of radiation. Mackenzie, furthermore, noted that there are specifications with respect to both the intensity of the UV-C radiation, and the wavelength of that radiation. Mackenzie notes that "a dose of 40mJ*cm^-2 is sufficient in terms of intensity, while two specific ranges of wavelength (264 nm & 207-222 nm) will attack two distinct characteristics of viruses [6]. The automated card dealing robot proposed herein will therefore design the UV-C-LEDs to emit these two frequency bands within the sterilization compartments. (AM)

There are projects that involve a similar concept to the proposed card shuffling and dealing robot. With that in mind, there are components that will be implement that these other projects do not have. An example of this kind of project is a robot that was developed called The Card Dealer. This robot runs using a keypad for user interfacing and uses an Arduino microcontroller to operate [7]. The proposed design does not include a keypad, as the proposed design will implement voice operated commands to reduce physical contact with the robot. Also, a PIC board will be used in lieu of an Arduino or any mass-produced microcontrollers. There will be absolutely no physical contact between the card players and the robot. The only physical contact will be between the card players and the cards. However, UV-C LEDs will be included that will clean every individual card each time the card deck is shuffled. This method of cleaning the cards is not included in the article that details a similar robot concept called The Card Dealer [7]. This cleaning process will keep the entire game of cards clean for each player, which is important since, during this time, sanitization is critical. The robot design will also need to include many different electrical components to get the design up and running. (DD)

These various electrical components will be required as demonstrated in the project by Arvid and Dariush [7]. In their project, they used three DC motors, two of which were stepper motors. For shuffling, they used a riffle shuffle method, which splits the deck in half and then places one card at a time, alternating from each half, into a space where the new deck will be. This was implemented by having a human manually split the deck in half. After the deck was split, two wheels were used, driven by a single DC motor and a gear system, to take one card at a time from each half and place it into a designated space. This, roughly, will be the approach taken in the automated card dealing robot proposed herein. In the project by Arvid and Dariush [7], the dealing was done with a stepper motor to rotate to the desired angle (where the player is seated), and then shot out by another motor which drives a wheel. The next issue, then, is to properly implement the shuffling of the decks of cards. (AM)

There are multiple different methods to shuffling playing cards. However, there is one method that stands out as the best option for the herein proposed case, the riffle shuffle. There are already playing card shuffling machines that are used in casinos. There is a type of card machine that is a “widely used machine that uses an ordinary riffle shuffle by pushing two halves of a single deck together using mechanical pressure” [8]. This method allows for randomness of the cards to come naturally with slight differences in alignment. The proposed design will not be using this method. Using this method would make it difficult to clean the cards individually with our UV-C LED circuit. There are also machines called shelf shufflers. Shelf shufflers have multiple shelves and randomly select different shelves to place cards, then randomly assembles the stacks [8]. Although this design achieves randomly shuffling the playing cards, this design is much too complicated for what is intended herein. The proposed design consists of a series of different compartments. The first compartment is where the cut decks will be placed. A dual motor, two-wheel system will be utilized which will place the bottom card from either of these two halves, in an alternating manner, into a new, shuffled compartment. While each card is sent into the shuffled compartment, it will be subjected to a UV-C LED circuit for card sterilization. This shuffling process is repeated until all cards within the shuffled compartment have been sterilized. For dealing, the newly shuffled deck will be placed into a new compartment. The cards will be dealt either face-up or face-down, depending upon which voice command is provided.aq (DD, AM)

There exists a patent for a card shuffling and dealing robot as well. This patent is for a robot that shuffles cards for any card game that has multiple rounds where cards need shuffled and dealt [9]. This design does not deliver cards to the players, however. It utilizes a second mechanical device to move the sets of playing cards from the shuffling 4apparatus to the playing card output area [9]. The herein proposed robot will be shuffling and dealing specifically for games of Blackjack. Cards will also be dealt to the player directly from the robot as the playing cards will be ejected directly to each individual player. This patent also does not include any card sterilization process or voice operated commands. (DD)

1.4. Marketing Requirements:

1. Ability to support multiple players while being autonomous
2. Sterilize playing cards
3. Ability to be plugged into a standard wall outlet
4. Ability to rotate to multiple player locations
5. Successfully shuffle a deck of cards
6. Ability to deal cards to players at different positions

(GC, DD, AM, JR)

2. Engineering Analysis

2.1. Circuits

1. Power Supply

The input to the power supply will be 120VAC. A modular power supply will be used to turn the 120VAC signal into either a 3.3VDC, 5VDC, or 12VDC voltage level. These DC voltage levels will then be regulated into one 30VDC voltage which will then route, in parallel to the input of every motor driver circuit. Another 8-15VDC voltage will be distributed to the microprocessor for input power. A 12VDC level will be sent to the sterilization circuit. The maximum power consumption of all of the subsystems combined in the worst case is roughly 200W, so the power supply should be at least 250W. (GC)

2. Motor Driver Circuits

The motor driver circuits will each intake a 10VDC to 30VDC. Each driver circuit will also be controlled in terms of time by the microprocessor. Since each motor will only operate when excited by an input voltage, the timing of each driver circuit being activated will be dealt primarily within the microprocessor. Each motor driver circuit will output the required voltage level for each individual motor. Although the max output of each driver circuit is 5VDC, this voltage level will likely need to be more finely tuned to achieve the correct torque output of each motor. Since each driver circuit will pull between of 0.2A to 1.0A, the total current draw of the motor driver circuits will be not greater than 5A. (AM)

3. UV-C Sanitization

The UV-C sterilization process will take place while each individual card is in transit from the shuffling compartment to the dealing compartment. The sterilization circuit will be comprised of 10 UV-C LEDs, each having a power consumption of 2.5W. The LED circuit will be connected in parallel in order to prevent one broken diode from breaking the circuit. An input voltage of 12VDC was selected such that the current draw of each diode is no greater than 210mA. Since the max current draw for each diode is 210mA, the total maximum current consumption of the sterilization circuit is 2.1A. An exposure time of 1 second is assumed while the card is traversing the sterilization chamber, although this time may vary depending upon the physical dimensions of the sterilization chamber. This exposure time will ensure proper sanitization of each playing card. (AM)

4. Keypad Subsystem

There will be 3 keypads each with 4 commands on to represent the 3 players and 1 keypad with 3 commands on it to represent the dealer. The commands that all controllers will have is “hit”, “stand”, and “bust”. The 4^{th} command on the 3 controllers will be a “Play Active” button that will be pressed to add to the total player count. For each command, a different combination of voltages will communicate to the microcontroller to perform the actions needed for that command. (JR)

2.2. Signal Processing

1. Speech Recognition

The speech recognition subsystem will be a speaker-independent speech recognition system and must recognize multiple commands from the human voice. These will come in as an analog wave form from a microphone that has a frequency range of 100 Hz to 10kHz to account for the range of human voice. This analog audio signal will then pass through an active audio amplification/filtering circuit, which will output a 5.5V-max voltage to the A/D converter input. The digitized signal will then be compared to sample waveforms of each spoken command. These will then be converted into a digital control signal that is sent to the microprocessor to control the driver circuits. This will no longer be implemented in the system and is being replaced with an individual keypad with buttons. (JR)

2. System intercommunication

The waveform that will be input to the processor will be a PWM signal from the analog to digital converter. The waveform that will be input into the processor will be a PWM signal from the analog to digital converter. This represents the audio that was given from the microphone and will be analyzed. Depending on the received output from the speech recognition functions, the processor will generate a new PWM signal for each driver circuit that is needed. This control signal will be different depending on what voice command is registered. After the signals are sent to the required driver circuits, there will be a pause to allow the process to complete before another voice command is accepted. (DD)

Due to changes is scope, the processor will be accepting input from a series of buttons that are on keypads instead of using voice commands. These will be wired directly to digital input pins. This means we will no longer need to use the analog to digital converter. The control/timing signal that will be generated from the processor will be a square waveform. Due to the nature of square waveforms, the PWM module of the processor will not need to be used. A series of logic high values and logic low values will be used to generate the necessary square waveforms. (DD)

2.3. Electromechanics

1. Motor Specifications

Each motor in the system can be defined in terms of its minimum required output torque. The equation which governs torque is:

$$T = Fd \qquad [1]$$

Where T is torque, F is force, and d is distance. The two shuffling motors have a required distance of travel of 6 inches. The dealing motors each have a minimum required distance of travel of 3 feet. The rotational dealing motor has a minimum required distance of travel of roughly 3.2 inches per rotation. To find the force required for each motor, the mass of the various components was assumed. The mass of each playing card was assumed to be 2 grams. The mass of the entire structure was assumed to be 4.5 kilograms. With all of these assumed masses, the required force can be found with the following equation:

$$F = kma \quad [2]$$

Where F is the force, k is an assumed coefficient of friction (assumed to be 0.5), m is mass, and a is acceleration downward due to gravity. The required torque for each component is as follows: 0.2 inch-ounces for the shuffling motors, 1.27 inch-ounces for the dealing motors, and 1.76 Newton-meters for the rotational motor. Four identical stepper motors were selected for the shuffling and dealing motors, which have a maximum output torque of 2.2 inch-ounces. The rotational motor selected has a maximum output torque of 1.99 Newton-meters. (AM)

2. Shuffling Motors

The minimum required output of the shuffling motors is 0.2 inch-ounces. This is the required torque to send an individual card forward by six inches. Since the selected stepper motors have a max torque output of 2.2 inch-ounces, they will be able to easily meet the minimum torque requirement. The shuffling motors will each take in a 4.8VDC max voltage level. The cards will be dispensed in an alternating manner through the sterilization chamber and into the dealing compartment. (AM)

3. Dealing Motors

The minimum required output of the dealing motors is 1.27 inch-ounces. This is the required torque to send a single card either face up or down, up to three feet (or thirty-six inches) forward to the required player. Since the selected stepper motors have a max torque output of 2.2 inch-ounces, they will be able to easily meet the minimum torque requirement. The shuffling motors will each take in a 4.8VDC max voltage level. The cards will be located in the dealing compartment after sterilization. The first dealing motor will distribute a face-down card forward to the player. This same motor will also distribute the face down card backwards into a chute at the same 1.27 inch-ounces. This will ensure that the card will make it down the chute since the torque is calculated to work for thirty-six inches; this chute will roughly be twelve inches travel distance. The second motor will distribute the card from the chute to the player. The final dealing motor will need a minimum torque of 1.7595 Newton-meters to rotate the entire system up to 180 degrees at 36-degree intervals to each player. (GC)

2.4. Embedded Systems

The Explorer 32 development kit that we will be using has the PIC24FJ128GA010 microprocessor which is a 16-bit microcontroller. The development kit we are using will require a DC voltage that is anywhere between 8VDC and 15VDC. This microprocessor runs at a maximum of 32 MHz and has 100 pins. This will have 8 kB of RAM and operated off of flash memory that is 64 kB of ROM. This microprocessor also needs to be within the temperature range of -40 degrees C and +85 degrees C in order to operate. The environment that we intend on using this microprocessor is going to be room temperature which will be well within this temperature range. This microprocessor has two ports, one for a maximum input signal voltage of 3.3VDC and one for a maximum input signal voltage of 5.5VDC. (DD)

3. Engineering Requirements Specification

Marketing Requirement	Engineering Requirement	Justification
3	Successfully operated by 120VAC wall power while using no more than 250 Watts**.**	120VAC is the standard output from a wall outlet in the USA, so this robot should be able to be run on that.
1	System will provide remote input capability for each user for autonomous operation.	To minimize handling of cards, the robot should be able to run fairly autonomously with minimum human interaction using various input commands.
5	Perform riffle shuffle one full deck of cards within 60 seconds	This is a reasonable amount of time to completely shuffle every other card in an entire deck of cards.
2	Sterilize a playing card within 1 second.	This amount of time is required in order to achieve a near 100% sterilization using UV-C radiation.
4,6	Rotate dealing enclosure in 36 degree increments to within +/- 5 degrees.	Since 4 players will be seated at the table, the robot should be able to rotate to each player within a reasonable degree of variation.
4,6	Should dispense a single card to a distance from 1 to 3 feet from opening of bot.	Since the players will be seated no more than 3 feet away from the robot, the card should be placed within arm’s reach of the player being dealt to.
1	The system should be able to respond to user input and provide visual feedback to alert the players within 2 seconds.	The system needs to provide convenient use by giving visual feedback to the players

		that a button command was accepted.
1	Register independent game flow signals from player keypads.	Players need the ability to make the game progress.
6	Should dispense a single card face up or face down.	Since blackjack requires both face up and face down cards to be dealt, the robot needs to be able to have that capability.
6	Operate with a maximum of 4 players	This is the max number of players required to play the game.
1. Ability to support multiple players while being autonomous **2.** Sterilize playing cards **3.** Ability to be plugged into a standard wall outlet **4.** Ability to rotate to multiple player locations **5.** Successfully shuffle a deck of cards **6.** Ability to deal cards to players at different positions		

(GC, DD, AM, JR)

4. Engineering Standards Specification

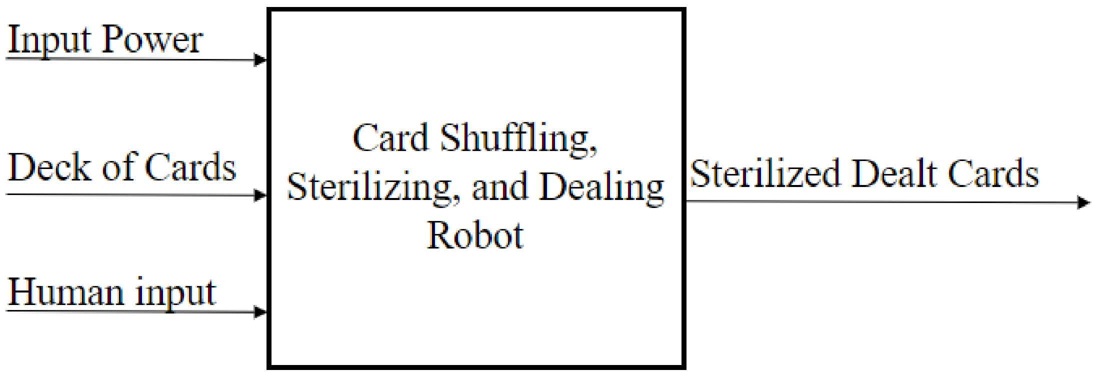
USB and Basic C are standards that we are looking into using/needing.

(GC, DD, AM, JR)

5. Accepted Technical Design

5.1. Hardware Design

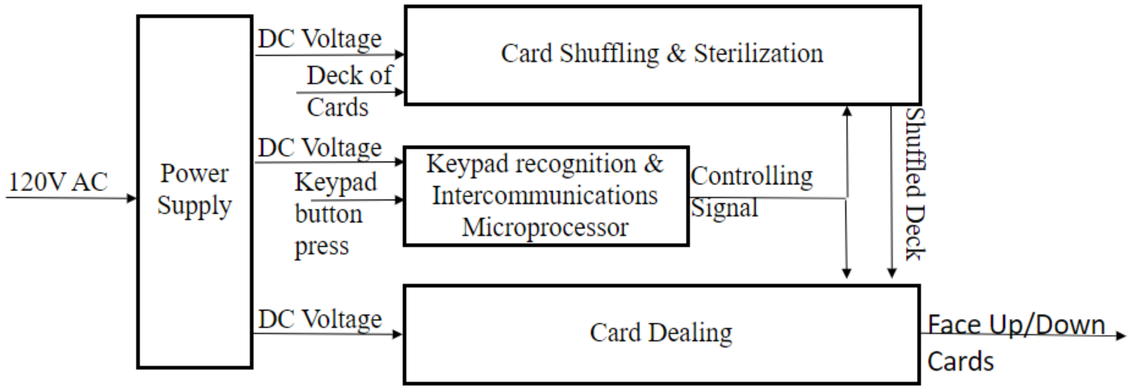
Level 0 Block Diagram:



Module	Card Shuffling, sterilizing, and Dealing Robot
Designer	Grant Crump, Drake Drivere, Adam Miner, Jakob Reiter
Inputs	Power: 120 VAC rms, 60 Hz Keypad button Cards
Outputs	Shuffled and sterilized cards
Description	Collect cards after a game has been played, shuffle and sterilize the cards using UV Light. Deal the correct number of cards to each player for the type of game selected. Recognize command based on what button was pressed on the keypad.

Functional Requirement Table: 1

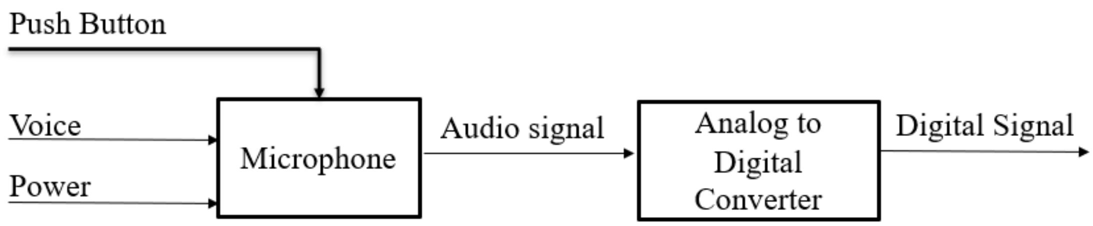
Level 1 Block Diagram:



Module	Level 1 Block Diagram
Designer	Adam Miner, Grant Crump, Drake Drivere, Jakob Reiter
Inputs	120V AC, Keypad button press, Two decks of cards
Outputs	Face Up or Down Cards
Description	Wall power, signal from keypad, and two decks of cards will be input into the system. The system will take the cards, shuffle them, sterilize them, and distribute them to the specific players turn face up/ down.

Functional Requirement Table: 2

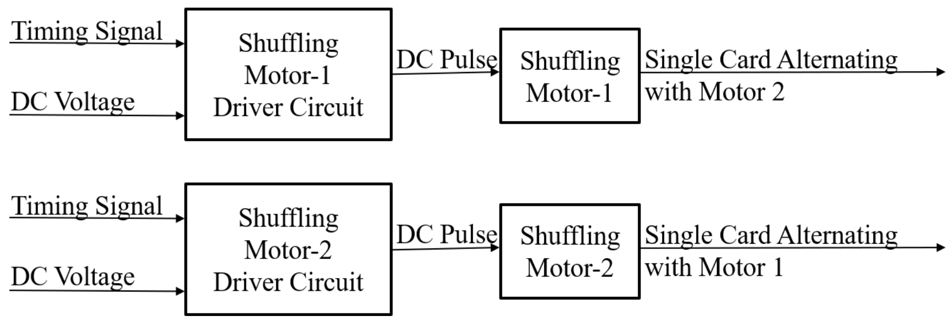
Speech Recognition Block Diagram:



Module	Speech Recognition
Designer	Jakob Reiter
Inputs	Push Button Human Voice Input Power
Outputs	Digital Signal to Microcontroller
Description	User presses a push button and speaks into the microphone saying a command. Command is taken into the audio to digital converter, converted to a digital signal and sent to the microcontroller.

Functional Requirement Table: 3

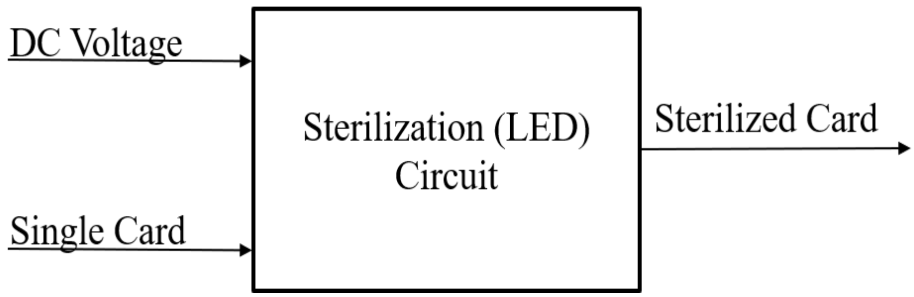
Original Card Shuffling Block Diagram:



Module	Card Shuffling
Designer	Adam Miner
Inputs	Timing Signal, 10DC Voltage, ½ Deck of Cards
Outputs	Single Card
Description	The shuffling motor driver circuits will receive both a 10VDC level and a timing signal to determine whether or not they should be engaged. Once the motors are engaged, they will dispense one card from the bottom of either side in an alternating manner into the sterilization chamber. The motors will be driven by a 4.8VDC signal, whose input voltage will be modified as needed in order to establish the required torque for the shuffling system.

Functional Requirement Table: 4

UV-C Sterilization Block Diagram:



Module	UV-C Sterilization
Designer	Grant Crump
Inputs	12VDC, Single Card
Outputs	Sterilized Card
Description	Using 10 UV-C LEDs, the sterilization chamber will use a 12VDC input to turn on all 10 LEDs during the shuffling process. Since each card will be passing through the chamber while the decks are being shuffled, the LED circuit will remain on throughout the duration of the shuffling process.

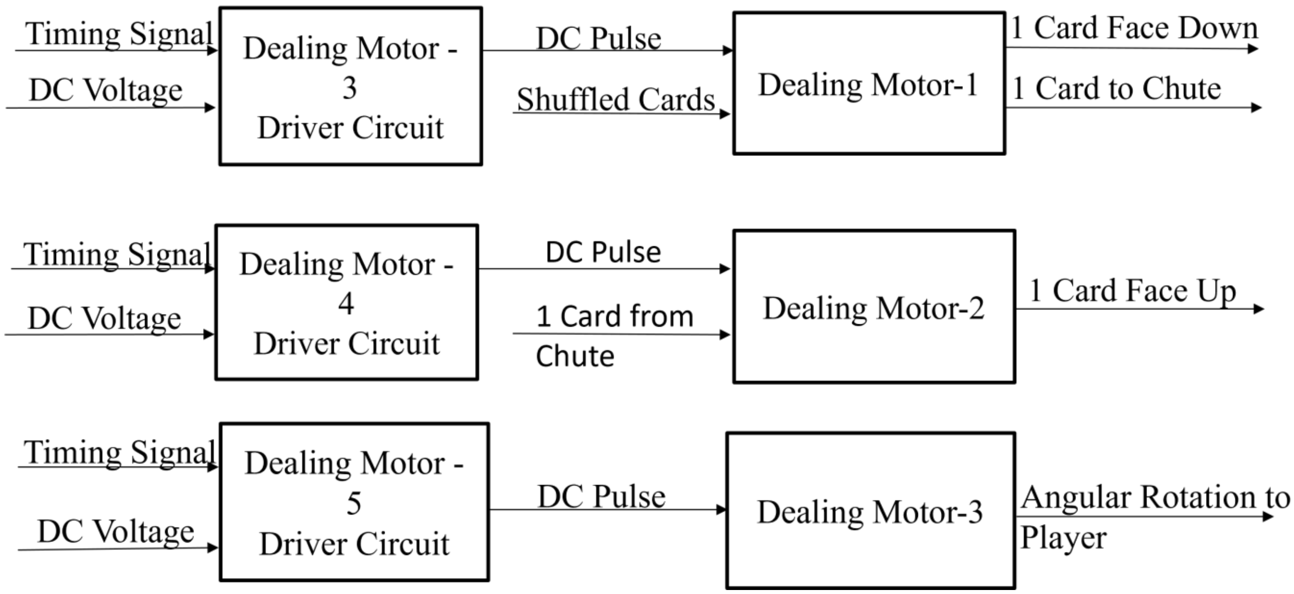
Functional Requirement Table: 5

Voltage Regulation Diagram:

Module	Voltage Regulation
Designer	Grant Crump
Inputs	30V Voltage
Outputs	DC Voltage: 12V, 10V, 5V, 3.3V
Description	30 VDC will be fed into the voltage regulation circuit. The circuit will then distribute the required 12VDC, 10VDC, 5VDC, and 3.3VDC voltage levels to all required subsystems.

Functional Requirement Table: 6

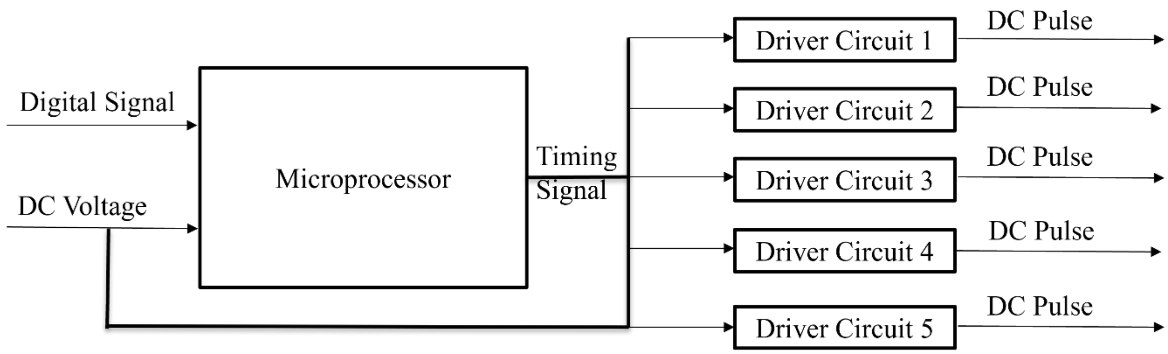
Card Dealing Block Diagram:



Module	Card Dealing
Designer	Adam Miner
Inputs	Timing signal DC voltage 10- 30 V
Outputs	Cards to players Angular rotation to players turns
Description	The driver circuits take in the voltage with a specific timing signal. This pulse goes into each dealing motor (1-3). Motor one will take in the shuffled cards, it will distribute one face down card to the player and one card to a chute to motor two. Motor two will distribute a card face-up to the player. For motors one and two, motor three will work simultaneously to rotate to which ever players turn it is.

Functional Requirement Table: 7

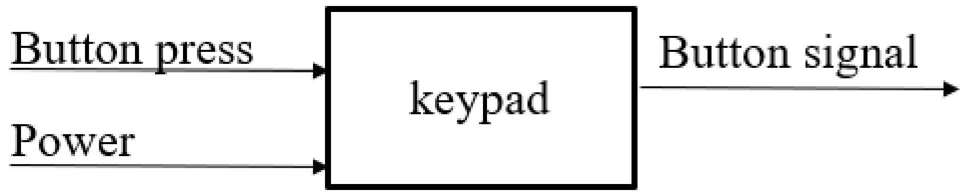
Inter-Communications Block Diagram:



Module	Inter-Communications
Designer	Drake Drivere
Inputs	Digital Signal, DC Voltage (8VDC – 15VDC)
Outputs	Driver Circuit 1-5 DC Pulse = 4.8VDC Max
Description	Square waveform timing signal generation based off the player input given to each driver circuit.

Functional Requirement Table: 8

Keypad Block Diagram:



Module	Keypad
Designer	Jakob Reiter, Adam Miner
Inputs	Power, Button press
Outputs	Digital signal
Description	User inputs on a keypad, which takes the signal necessary to do the certain command and outputs as a digital signal. Commands: Hit, Stand, Bust, Player count

Functional Requirement Table: 9

Circuit Schematics:

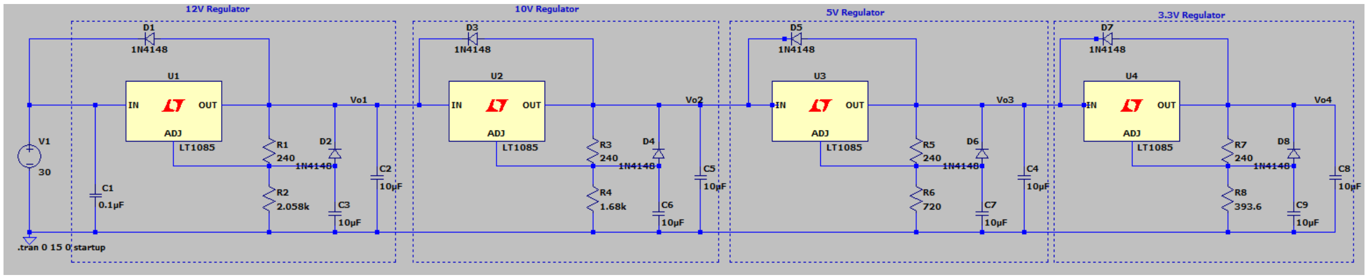


Figure 1: Voltage Regulation Circuit

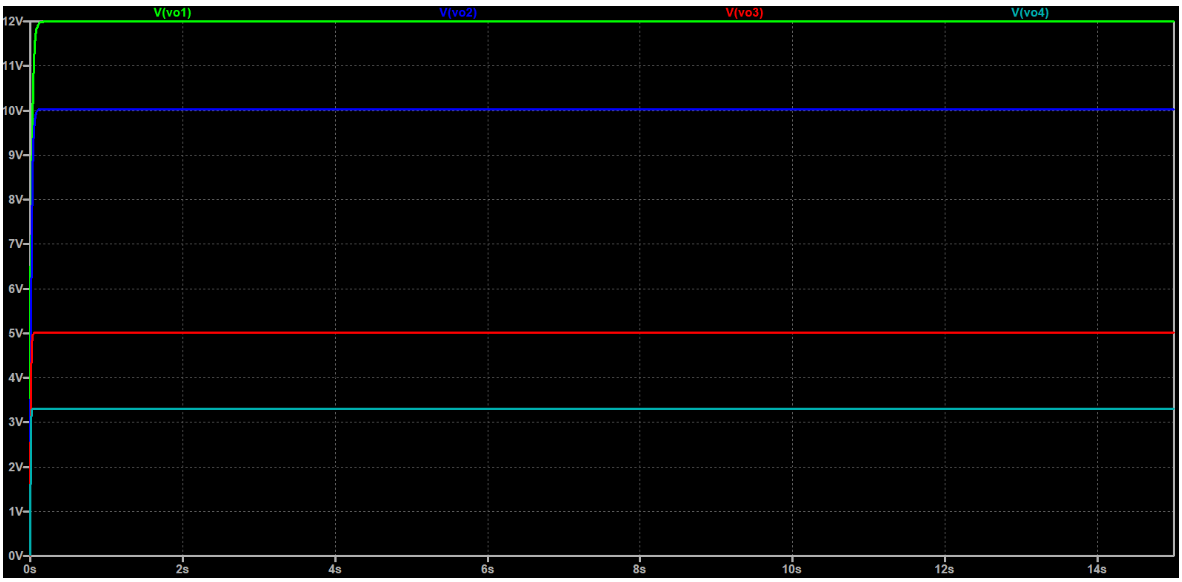


Figure 2: Voltage Regulation Output Waveforms

$$Vout = VREF\left(1 + \frac{R2}{R1}\right) \qquad [3]$$

The voltage regulation as seen above in figure 1 shows a circuit that will input a voltage of 30 V DC, and for each section in parallel there is a 12V, 10V, 5V, and 3.3V regulator section from left to right. Each section is constructed similarly with two diodes, three capacitors, two resistors and one chip. The diodes are for input short circuit (D1) and output short circuit protection (D2). Capacitors C3 and C2 for example are used for ripple rejection up to 15 Db and helps improve transient response respectively. Resistors R1 and R2 for example are used for limiting the output current that the regulator can provide to the load. For each regulator R2, R4, R6, and R8 were found using the equation 3 above based on the required output voltage. The chip LT1085 is a positive adjustable regulator. VREF in the equation above is 1.25V and this value is the voltage the chip outputs across the OUT and ADJ pins. Each regulator section was constructed the same aside from the resistors R2, R4, R6, and R8 which were hand calculated as mentioned before. Figure 2 shows the output at each output node for the individual voltage regulation sections. (GC)

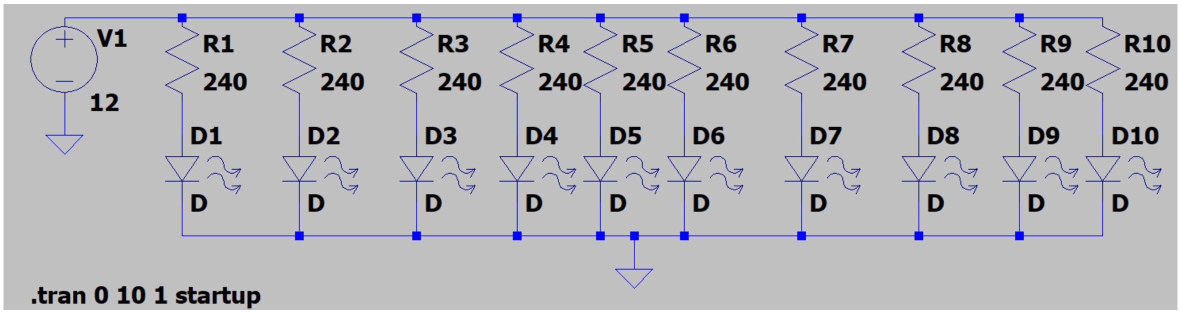


Figure 3: UVC-Circuit Initial Design

Figure 3, shown above, is the UVC-Circuit schematic. This circuit takes in 12 V DC which is in parallel to the 10 resistors and 10 LEDs. The resistors R1-10 are put in place to restrict the current flow into the LEDs, so they do not take in too much current and burn out. The LEDs are there to represent the UVC lights that would be used to sterilize the playing cards in the machine. (GC)

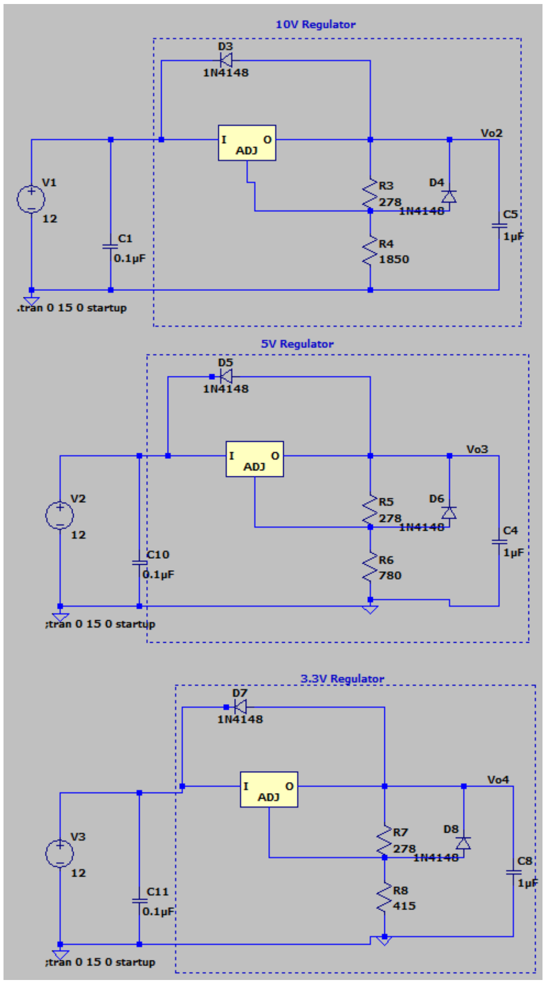


Figure 4: Voltage Regulation Circuit Final Implementation

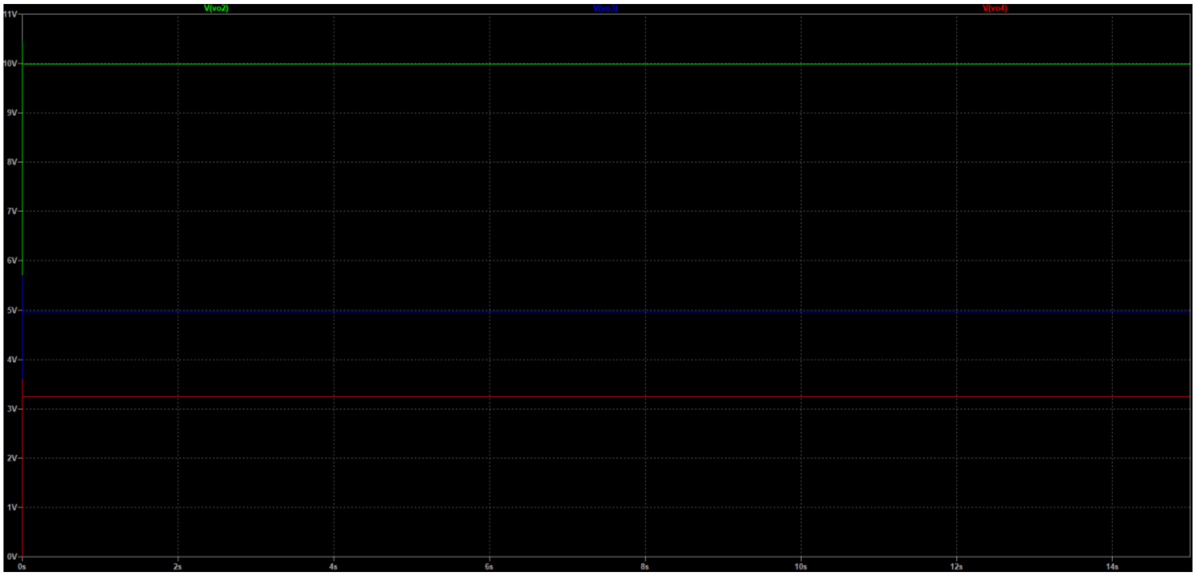


Figure 5: Voltage Regulation Output Waveforms Final Implementation

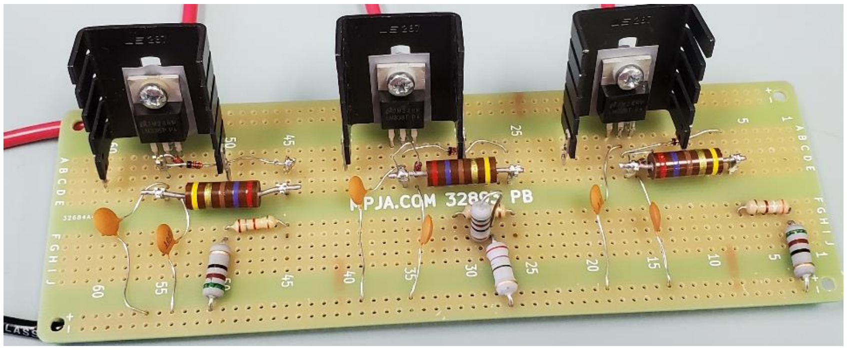


Figure 6: Voltage Regulation Physical Implementation

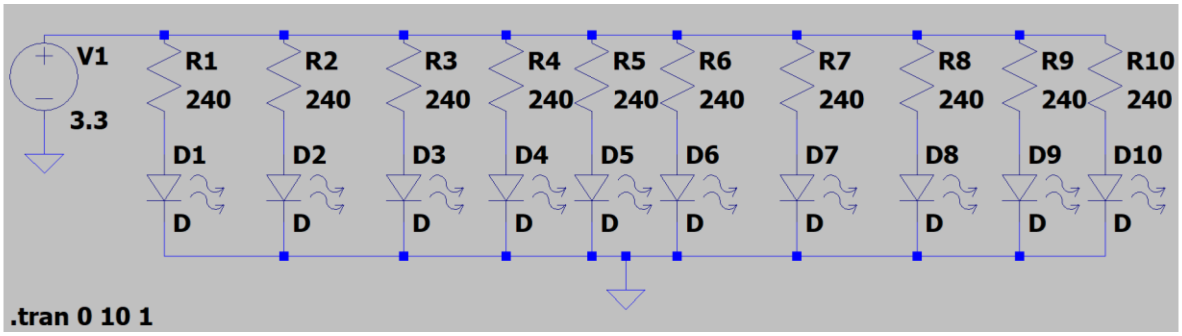


Figure 7: LED Circuit Final Implementation

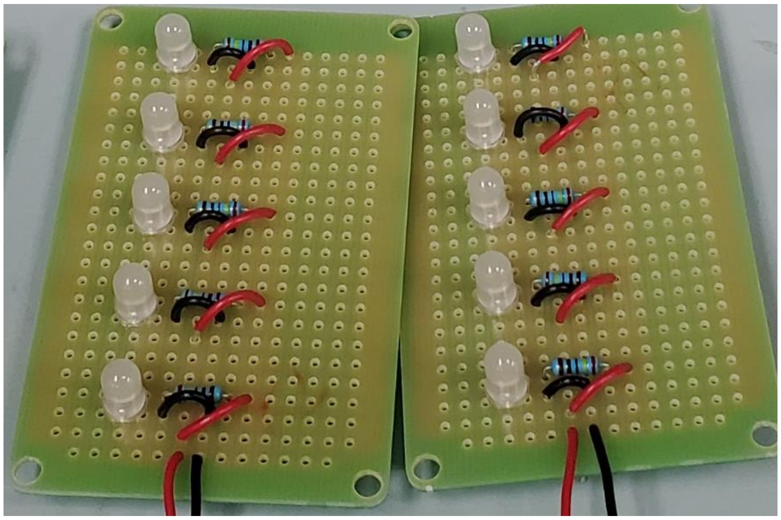


Figure 8: LED Circuit Physical Implementation

Figures 4-8 above are the final variations and implementations of the Voltage Regulation and LED circuits. The voltage regulation circuit was slightly changed from the previous iteration. The 30V DC input was changed to a 12V DC input to reduce the circuit size. The circuits were separated and connected to the 12V DC input in parallel instead of in series. This new iteration removed one capacitor from each circuit due to a loading issue when a load was applied to each of them. The use of a LM388T voltage regulation chip was used which is rated for 12V DC input and up to a 5A output. When the project was connected physically the need to use the 10V DC voltage regulation sequence was used to power the Explorer 16/32 board along with 3 DC motors. The LED circuit was changed from 12V DC a 3.3V DC input which was distributed from the Explorer 16/32 board. (GC)

The voltage regulation and LED circuits were edited

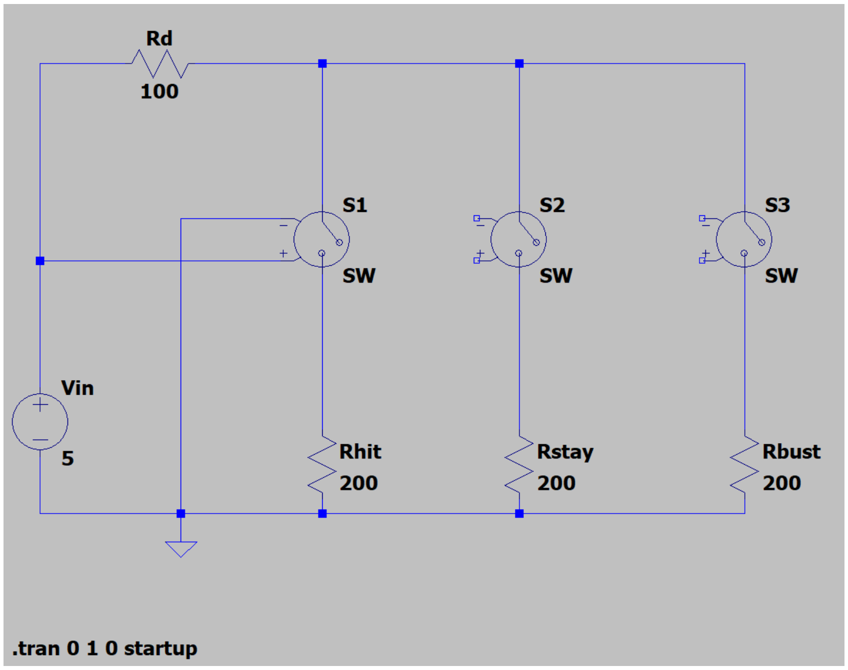


Figure 9: Remote Controller Circuit

Figure 9, shown above, shows the remote controller circuit for the hit, stay, and bust commands. Once the remote is activated via the microprocessor to indicate it is the player's turn, the player can press one of three possible buttons. This button will close a switch, which will send a 3.3 VDC signal to a pin on the microcontroller which will be associated with that command. For each button that is not pressed, a 0 VDC signal will be sent to the associated pin on the microcontroller to indicate that that command has not been sent. Another resistor, "Ractive", will be implemented in the same manner as the switch/resistor combinations above to designate whether or not a player is active. (AM)

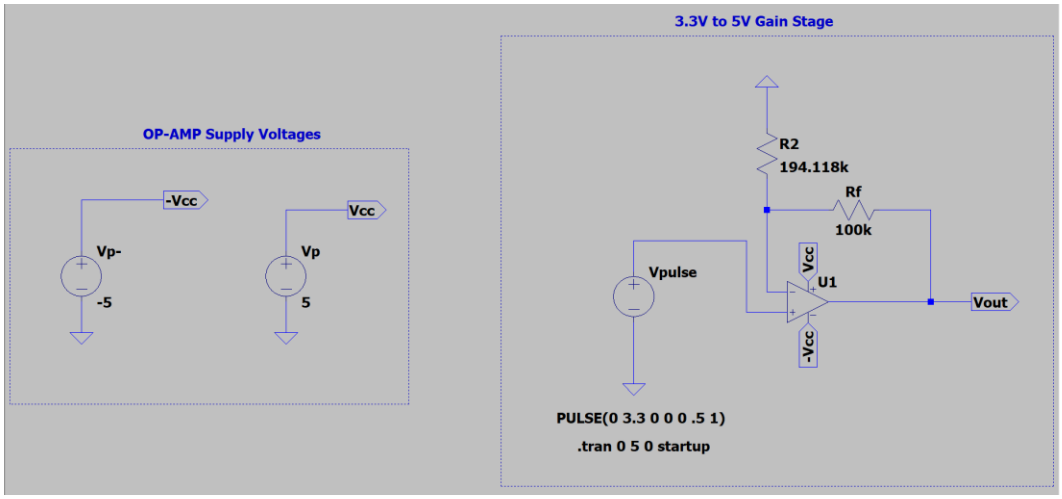


Figure 10: 3.3V to 5V Gain Stage

Figure 10 above shows the 3.3V to 5V gain stage for the timing/stepping signal that will be sent from the microcontroller to each motor driver circuit. Since each motor driver chip requires the controlling inputs to be either 0 V or 5 V, this gain stage is necessary. The output signal of the microcontroller will be a 3.3 V pulse signal. This signal will be amplified by a non-inverting operational amplifier with a gain of 1.5151. This therefore means that the motor driver chips will be receiving either a 5 V or 0 V signal, which is exactly what they require. (AM)

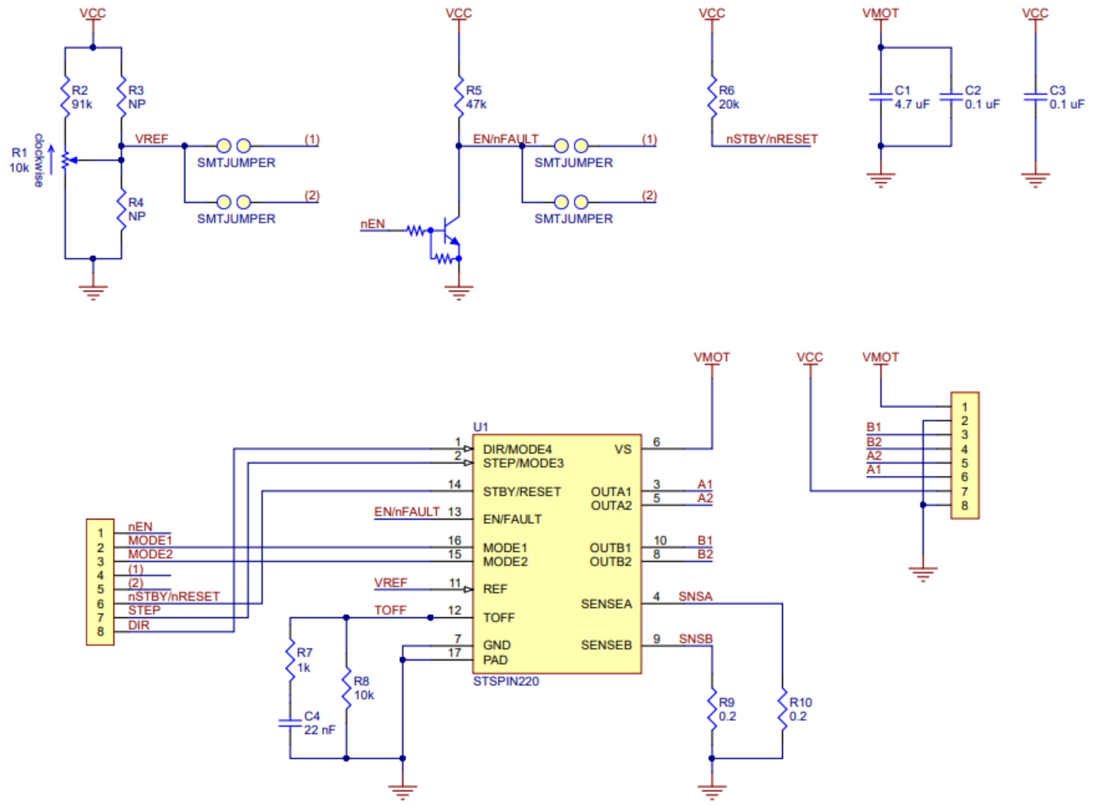


Figure 11: Motor Driver Schematic

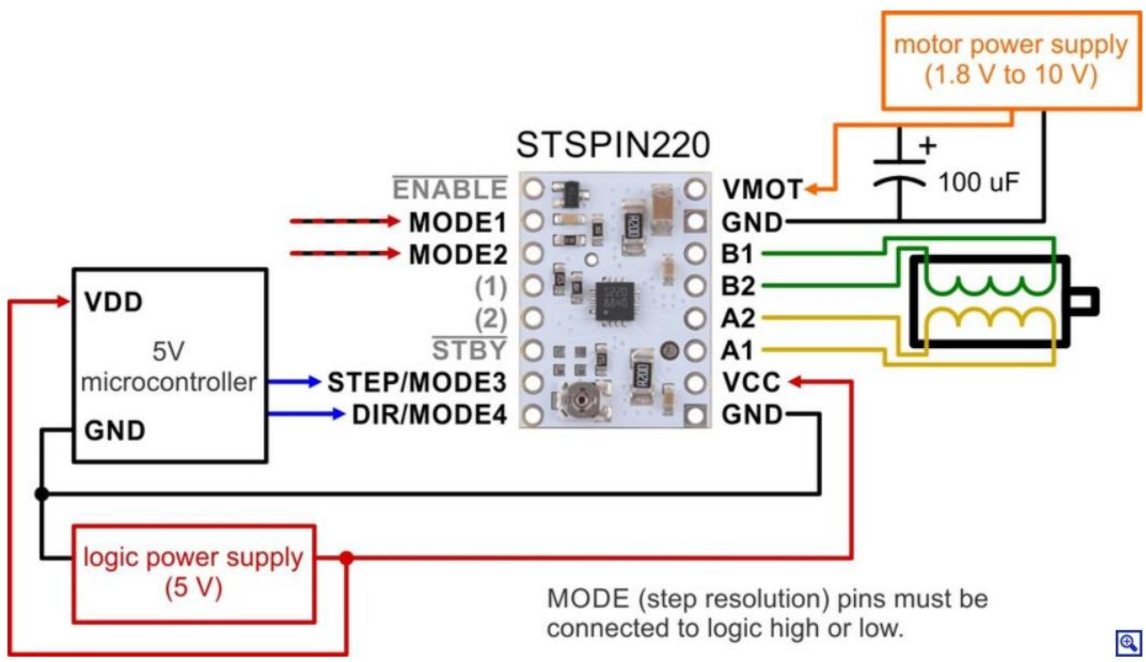


Figure 12: Motor Driver Wiring Diagram

Figure 12 above shows the most detailed schematic available for the motor driver circuit that will be used for the project. The driver circuit VDD pin is tied to 5 VDC, and

GND is tied to common ground. The VMOT pin is tied to 3.8 VDC for the rotational motor and 2 VDC for the two shuffling motors and the two dealing motors. Pins B1, B2, A2, and A1 are tied to each motor's input wires. Pins MODE1 and MODE2 will both be tied to common ground. STEP/MODE3 and DIR/MODE4 will be connected to the microcontroller pulse output. (AM)

Since the motors will be using their full step functionality, the step/direction pins on the driver circuit will be the only pins in use for the actual digital control of the motors. The two mode pins that will be tied to ground would only be non-ground if the system required fractional steps of the motors, which it does not. For the stepping pin, a controlling signal will be sent to the driver to take a number of steps depending upon which motor is in use. The dealing motor which sends the face-up card will also be utilizing the directional pin to send a card backwards. The rotational motor will also utilize the directional pin to rotate both forward and backward. (AM)

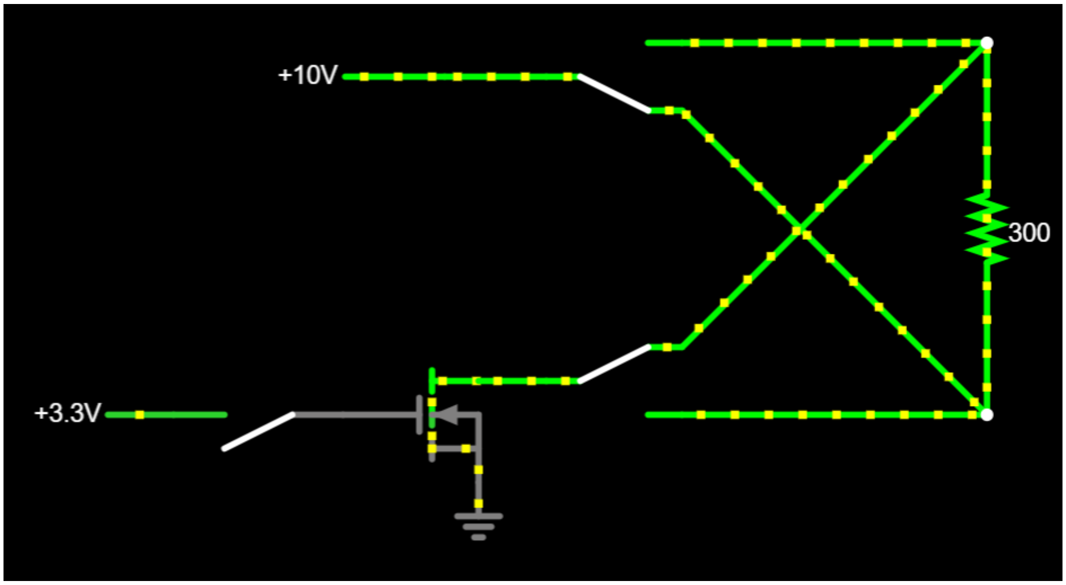


Figure 13: DC Motor Driver Circuit

The circuit shown above in Figure 13 was used to implement the bi-directional dealing motor. This was due to multiple failures of the original dealing stepper motors both with respect to output torque (the stepper motors did not have enough torque) and physical component failures within either the stepper motors or the drivers. A modification of this circuit (seen below in Figure 14) where the motor connects straight from the drain of the transistor to the 10 Volt source was used for the uni-directional dealing motor and one of the shuffling motors. The gate on both of these circuits' NMOS transistors was controlled by way of 3.3 Volt signal voltage from the microprocessor. (AM)

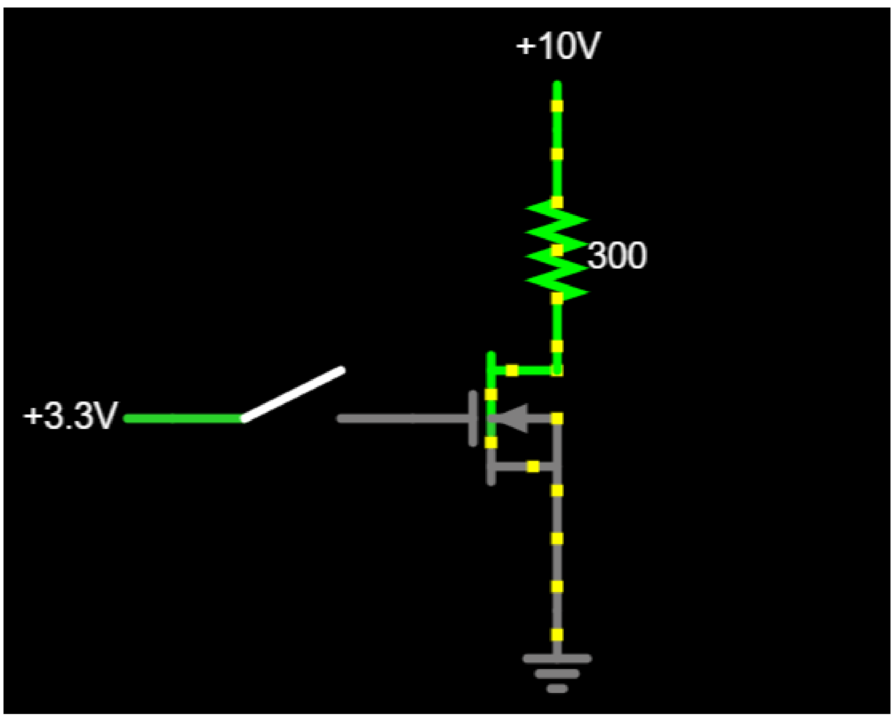


Figure 14: DC Motor Driver Circuit (Uni-Directional Modification)

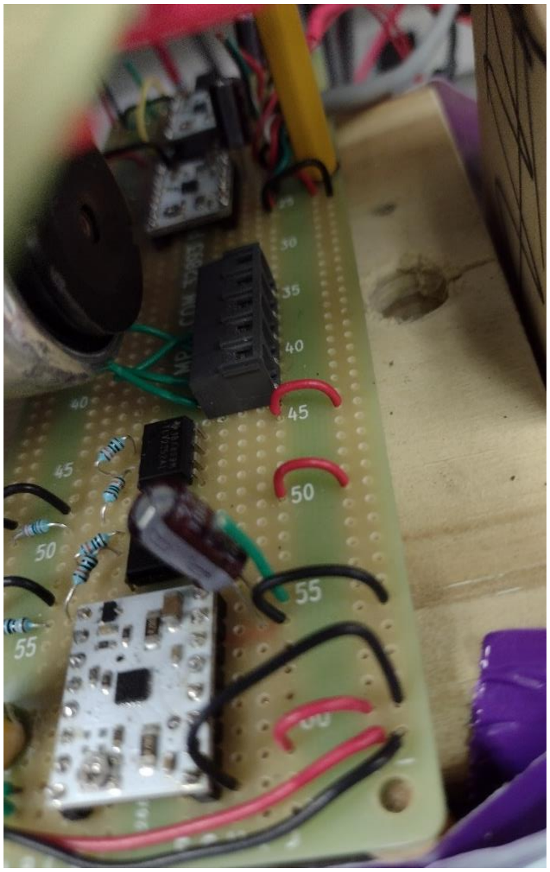


Figure 15: Actual Implementation of Stepper Motor Driver Circuit

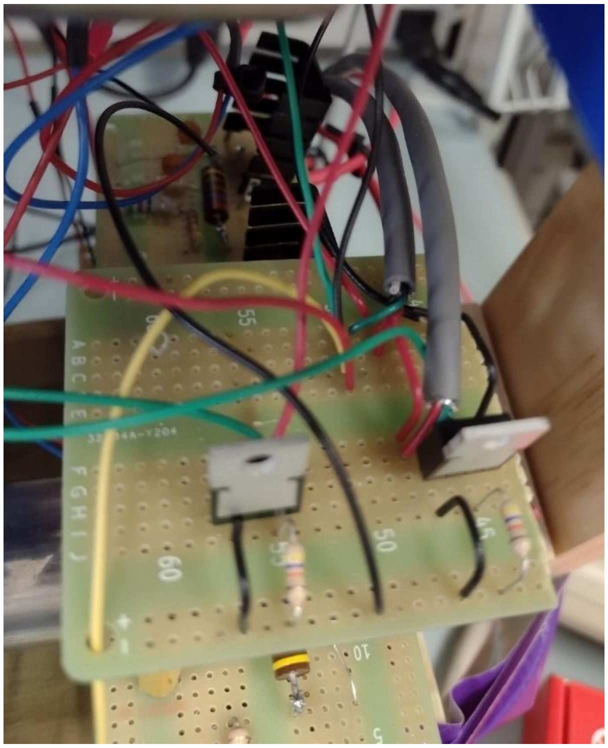


Figure 16: Actual Implementation of DC Motor Driver Circuit

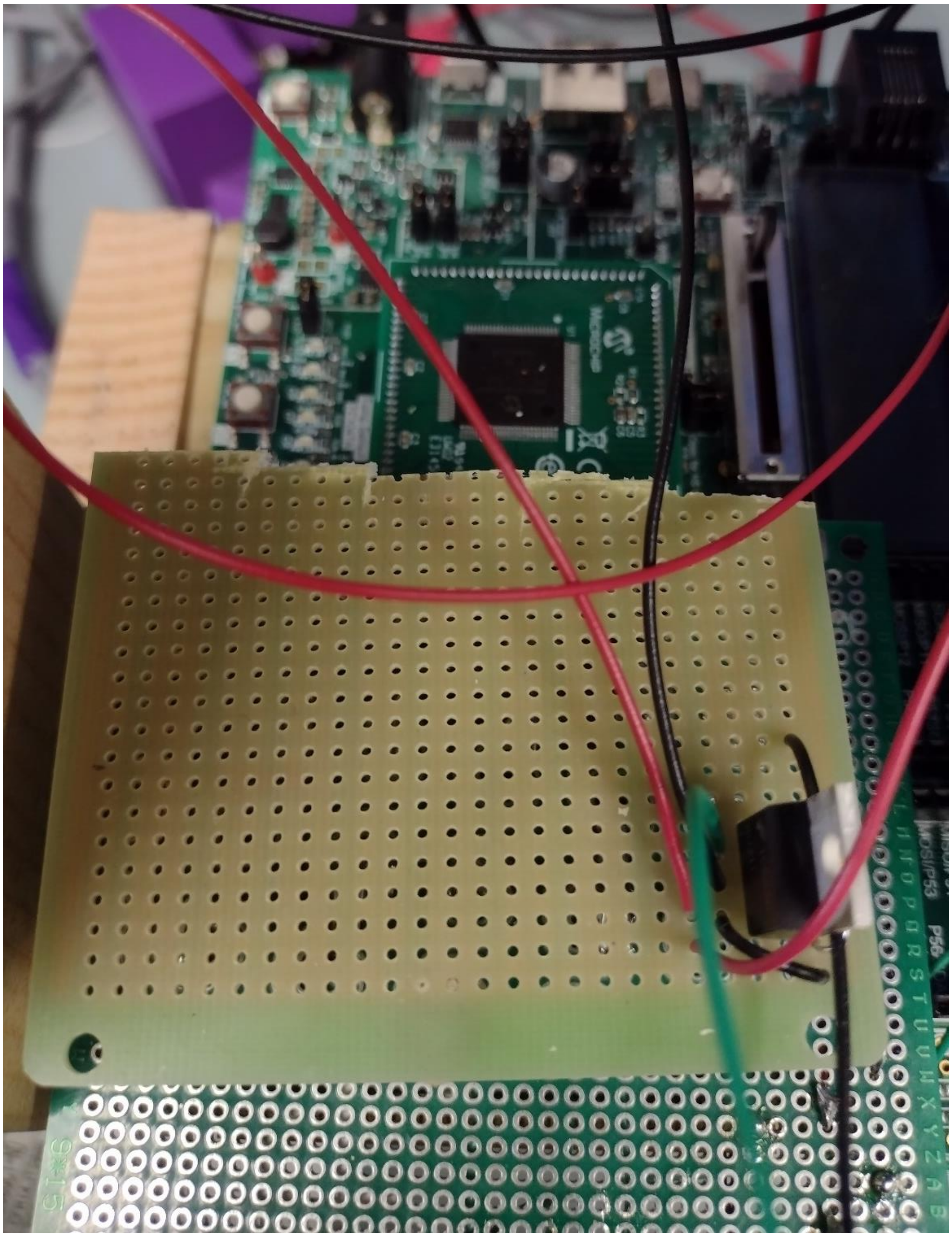
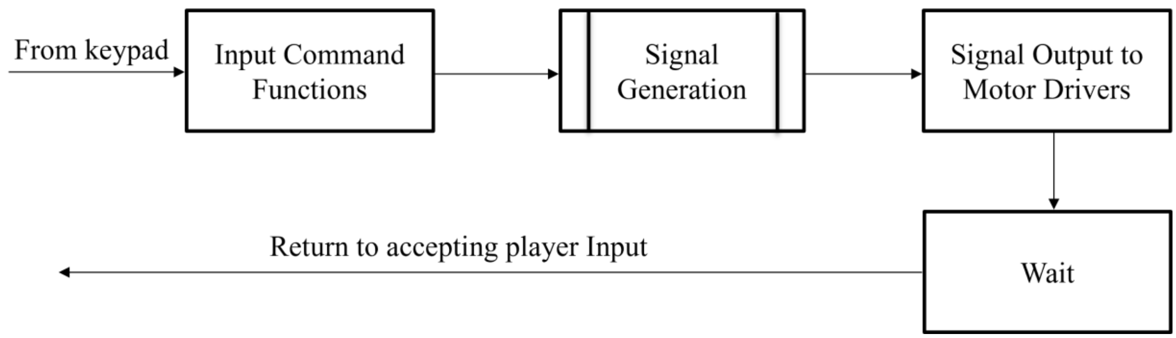


Figure 17: Actual Implementation of DC Motor Driver Circuit (Uni-Directional Modification)

Figures 15, 16, and 17 shown above show the actual implementations of the aforementioned circuits. Since 3 motor driver circuits failed, 3 stepper motors were replaced by 3 DC motors. The new DC motors were driven by the DC motor drivers shown above. (AM)

5.2. Software Design

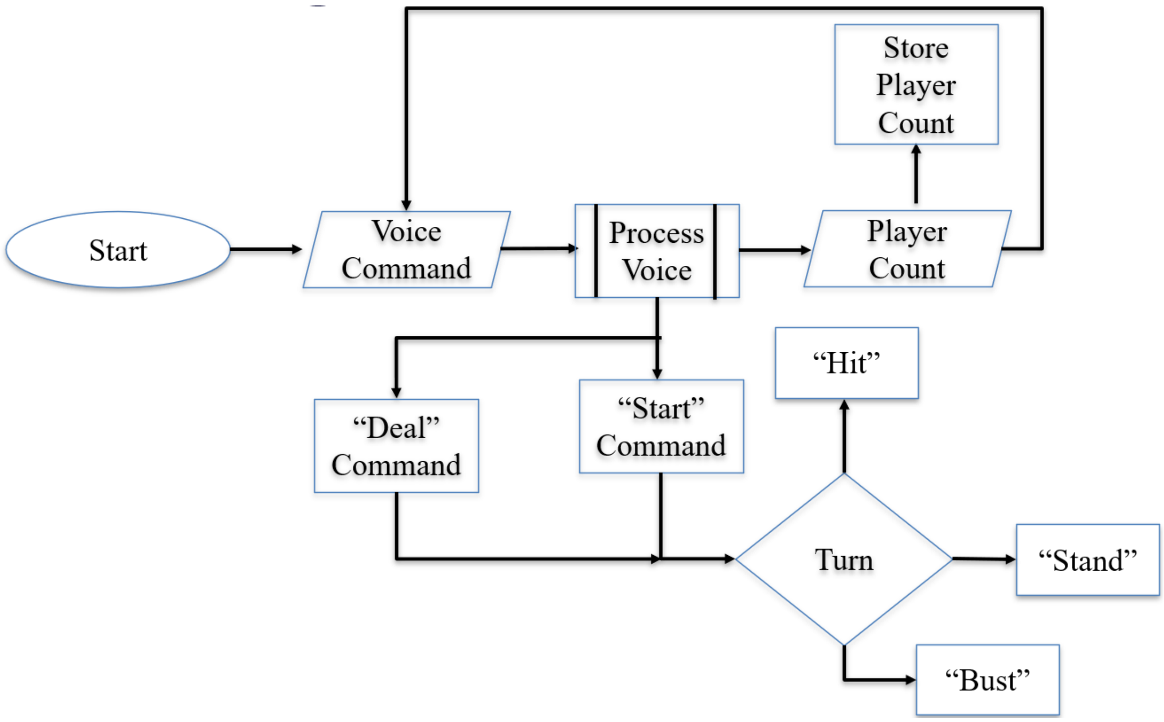
Inter-communications Flowchart:



Module	Inter-Communications Flowchart
Designer	Drake Drivere
Inputs	Speech Recognition Functions
Outputs	Return to player input system
Description	Square waveform timing signal generation that is dependent on keypad commands and the code loops back to accepting another command.

Functional Requirement Table: 10

Speech Recognition Flowchart:

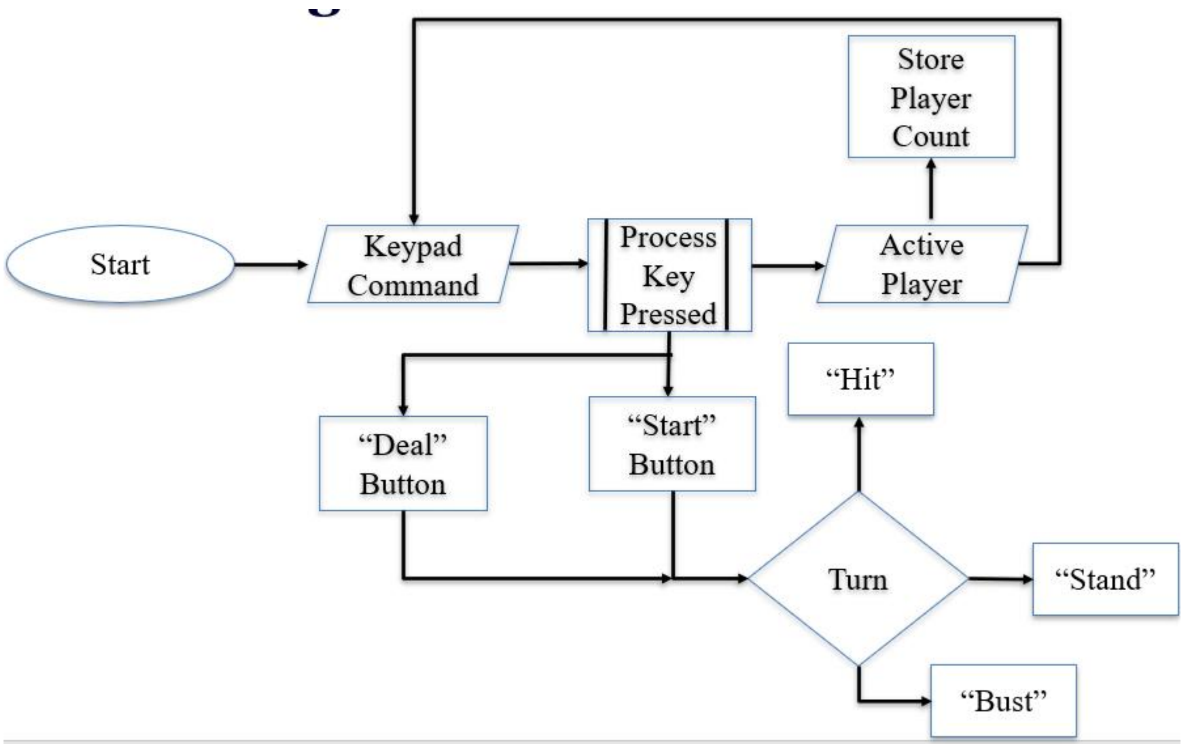


Module	Speech Recognition
Designer	Jakob Reiter
Inputs	Human voice
Outputs	Function call to each command
Description	Flowchart that describes the code loop of the speech recognition system. Takes the voice command, processes it and returns the function call for each command. "Hit" – System deals a card to current player. "Stand" – System moves on to the next player or, if there are no players remaining and waits for another command. "Bust" – Same as stand. Player count "1", "2", or "3" – There is 1 player automatically (the dealer) plus the number of players that is said for a total of 1 + 3 = 4 maximum players. "Deal" – Deal the initial turn of cards (1 face up, 1 face down) to

	everyone, and go back to player 1. • "Start" – Does what "Deal" does along with shuffling the cards.

Functional Requirement Table: 11

Keypad Flowchart:



Module	Keypad
Designer	Jakob Reiter
Inputs	Button Presses
Outputs	Function call to each command
Description	Flowchart that describes the code loop of the speech recognition system. Takes the button press command, processes it and returns the function call for each command. "Hit" – System deals a card to current player. "Stand" – System moves on to the next player or, if there are no players remaining and waits for another command. "Bust" – Same as stand. Active player – Dealer controller automatically is counted towards

	player count, the rest of the controllers will have a button that will add up the player count. "Deal" – Deal the initial turn of cards (1 face up, 1 face down) to everyone, and go back to player 1. • "Start" – Does what "Deal" does along with shuffling the cards.

Functional Requirement Table: 12

Code/Pseudocode:

This code demonstrates taking input from a button press and outputting a square waveform that will be a timing/stepping signal for the motor driver circuits. This code works with the on-board buttons; however, the physical implementation with system integration will be with external buttons. Using external buttons will leave the code unchanged. The code below is modular in nature which makes it trivial to add more input buttons and more output waveforms. A function of the code is a millisecond delay function which is used to generate the waveforms. This will allow for changes in the duty cycles of the waveforms. With a basic counter variable in the code, we can easily manipulate how long the timing/stepping signal is active. This gives all of the functionality that is needed for the motor driver circuits. (DD)

```
/**
 * Drake Drivere
 * This is working code that runs on the microprocessor that uses two temporary on-board
 * buttons that both generate and output individual square waveforms.
**/


#include "mcc_generated_files/system.h"
```

```
void ms_delay (int ms)
{
   T2CON = 0x8030; // Timer 2 on, TCKPS<1,0> = 11 this 1:256
   TMR2 = 0;
   while(TMR2 < ms*62.5);
}

int main(void)
{
   SYSTEM_Initialize(); //Function that is included in system file to initialize the board
   unsigned long i;
   int counter = 0;

   TRISA = 0xFF00; //set all PORTA pins as outputs
   TRISB = 0xFF00; //set all PORTB pins as outputs
   TRISD = 0xFFFF; //set all PORTD pins as inputs

   //loop that runs forever
   while(1)
   {
      //This code takes input from the on-board buttons and outputs a square waveform
      //accordingly
      if(PORTDbits.RD13 == 0) //ON-BOARD S4 button
      {
         //Runs the square wave for a given amount of time
         while(counter < 200)
         {
            //Runs square wave on portB(BIT 0) pin 25
            ms_delay(5);
            PORTB = 1;
            ms_delay(5);
```

```
            PORTB = 0;
            counter = counter + 1;
        }
        counter = 0;
    }
    else if(PORTDbits.RD6 == 0) //ON-BOARD S3 button
    {
        //Runs the square wave for a given amount of time
        while(counter < 1)
        {
            //Runs square wave on portB(BIT 1) pin 24
            ms_delay(5);
            PORTB = 2;
            ms_delay(5);
            PORTB = 0;
            counter = counter + 1;
        }
        counter = 0;
    }
  }
  return 1;
}
```

This is the code/pseudocode for the keypad logic. It uses the ports to see whether a button is pressed and depending on what button is pressed it will do a certain action.

```
while(1)
    {
        if(PORTB = 1)//player 1
        {
            //player active button is pressed
```

```
    //LED on controller will light up
    //This will increase a counter to get the total player count
}
else if(PORTB = 2 && playercount == 2)
{
    //hit command
    //deals one card to player 1
}
else if(PORTB = 3 && playercount == 2)
{
    //stand command
    //moves on to player 2 by rotating the system
    //player one is still in the game
}
else if(PORTB = 4 && playercount == 2)
{
    //bust command
    //player 1 went over 21 and is now out of the game
    //rotates the system to player 2
}
if(PORTB = 5 && playercount == 2)//player 2
{
    //player active button is pressed
    //LED on controller will light up
    //This will increase a counter to get the total player count
}
else if(PORTB = 6 && playercount == 2)
{
    //hit command
    //deals one card to player 2
}
```

```
else if(PORTB = 7 && playercount == 2)
{
   //stand command
   //moves on to player 3 by rotating the system
   //player 2 is still in the game
}
else if(PORTB = 8 && playercount == 2)
{
   //bust command
   //player 2 went over 21 and is now out of the game
   //rotates the system to player 3
}
if(PORTB = 9 && playercount == 3)//player 3
{
   //player active button is pressed
   //LED on controller will light up
   //This will increase a counter to get the total player count
}
else if(PORTB = 10 && playercount == 3)
{
   //hit command
   //deals one card to player 3
}
else if(PORTB = 11 && playercount == 3)
{
   //stand command
   //moves on to player 4(dealer) by rotating the system
   //player 3 is still in the game
}
else if(PORTB = 12 && playercount == 3)
{
```

```
        //bust command
        //player 3 went over 21 and is now out of the game
        //rotates the system to player 4(dealer)
    }
    if(PORTB = 13)//player 4//player 4 is dealer therefore always active
    {
        //player 4 is dealer therefore always active
        //hit command
        //deals one card to player 4
    }
    else if(PORTB = 14)
    {
        //stand command
        //moves back to original postion by rotating the system
        //player 4 is still in the game
    }
    else if(PORTB = 15)
    {
        //bust command
        //player 4 went over 21 and is now out of the game
        //rotates the system back to player 1
    }
}
```

(JR)

Finalized Code:

The finalized code needed to make some adjustments due to how the hardware was setup. The external button circuits were not hooked up, so the on-board buttons were used. All motors were initially stepper motors, however three of them were replaced with dc motors. The change in the code to implement the three dc motors was simple. This involved making the timing signals a constant high signal when the motor is running, while the stepper motors run with a square wave signal. The code also has the game logic for the game of Blackjack implemented. (DD)

```
/**
 * Drake Drivere, Jakob Reiter
**/

#include "mcc_generated_files/system.h"

void ms_delay (double ms)
{
    T2CON = 0x8030; // Timer 2 on, TCKPS<1,0> = 11 this 1:256
    TMR2 = 0;
    while(TMR2 < ms*62.5);
}

void dealFaceDown ()
{
    int counter1 = 0;
    while(counter1 < 250) // runs top dealing motor
    {
        PORTC = 16;
        ms_delay(1.5);
        counter1 = counter1 + 1;
    }
    PORTC = 0;
    counter1 = 0;
}

void dealFaceUp ()
{
    dealFaceDown(); // runs top dealing motor

    int counter2 = 0;
    while(counter2 < 250) // runs bottom dealing motor
```

```
    {
        PORTE = 128;
        ms_delay(1.5);
        counter2 = counter2 + 1;
    }
    PORTE = 0;
    counter2 = 0;

}

void hit ()
{
    dealFaceUp ();
}

void inverseRotate ()
{
    int counter1 = 0;
    while(counter1 < 50) // runs rotational motor
    {
        ms_delay(3.33);
        PORTE = 64;
        ms_delay(3.33);
        PORTE = 0;
        counter1 = counter1 + 1;
    }
    counter1 = 0;
}

void rotate ()
{
    int counter1 = 0;
    while(counter1 < 50) // runs rotational motor
    {
        ms_delay(3.33);
        PORTE = 80;
        ms_delay(3.33);
        PORTE = 16;
        counter1 = counter1 + 1;
    }
    counter1 = 0;
}

void shuffle ()
{
    int counter1 = 0;
    int counter2 = 0;
    int counter3 = 0;
```

```
    while(counter1 < 4) // runs shuffling stage
    {
        PORTE = 32; // this turns on the LEDS
        while(counter2 < 500) // runs right shuffling motor
        {
            ms_delay(1.5);
            PORTA = 1;
            ms_delay(1.5);
            PORTA = 0;
            counter2 = counter2 + 1;
        }
        counter2 = 0;

        while(counter3 < 250) // runs left shuffling motor
        {
            PORTA = 2;
            ms_delay(1.5);
            counter3 = counter3 + 1;
        }
        PORTA = 0;
        counter3 = 0;

        counter1 = counter1 + 1;
    }
    PORTE = 0;
    counter1 = 0;
}

int startGame (int playerCount, int maxTurn)
{

    int counter2 = 0;
    while(counter2 < 5333) // exercise bottom dealing motor
    {
        PORTE = 128;
        ms_delay(1.5);
        counter2 = counter2 + 1;
    }
    PORTE = 0;
    counter2 = 0;

    ms_delay(1000);
    ms_delay(1000);
    ms_delay(1000);


    int counter5 = 0;
```

```
while(counter5 < 5333) // exercise top dealing motor
{
    PORTC = 16;
    ms_delay(1.5);
    counter5 = counter5 + 1;
}
PORTC = 0;
counter5 = 0;

ms_delay(1000);
ms_delay(1000);
ms_delay(1000);

int counter3 = 0;
while(counter3 < 5333) // exercise left shuffling motor
{
    PORTA = 2;
    ms_delay(1.5);
    counter3 = counter3 + 1;
}
PORTA = 0;


// rotate system to default position and shuffle cards (WIP)
shuffle ();

if(playerCount == 1) //1p
{
    // system does not move
    // deal to just player 1
    dealFaceDown ();
    ms_delay(1000);
    dealFaceUp();
    maxTurn = 1;

    return maxTurn;
}
else if(playerCount == 2) //2p
{
    //deal to player 1 rotate to position 2 and deal to player 2
    dealFaceDown ();
    rotate ();
    dealFaceDown ();
    ms_delay(1000);
    dealFaceUp ();
    inverseRotate ();
    dealFaceUp ();
    maxTurn = 2;
```

```
        return maxTurn;
    }
    else if(playerCount == 3) //3p
    {
        //deal to player 1, 2 and 3
        //deal to 1 then rotates X degrees to 2 position then rotate to 3
position then returns to position 2
        dealFaceDown ();
        rotate ();
        dealFaceDown ();
        rotate ();
        dealFaceDown ();
        ms_delay(1000);
        dealFaceUp ();
        inverseRotate ();
        dealFaceUp ();
        inverseRotate ();
        dealFaceUp ();

        maxTurn = 3;

        return maxTurn;
    }
    else if(playerCount == 4) //4p
    {
        //deal to player 1, 2, 3 and 4
        dealFaceDown ();
        rotate ();
        dealFaceDown ();
        rotate ();
        dealFaceDown ();
        rotate ();
        dealFaceDown ();
        ms_delay(1000);
        dealFaceUp ();
        inverseRotate ();
        dealFaceUp ();
        inverseRotate ();
        dealFaceUp ();
        inverseRotate ();
        dealFaceUp ();
        maxTurn = 4;

        return maxTurn;
    }

}
```

```
int main (void)
{
    SYSTEM_Initialize();

    unsigned long i;
    int playerCount;
    int currentTurn;
    int maxTurn = 0;
    int turnFlag = 1;

    TRISA = 0xFFFC; // set portA to input except for a few pins

    TRISC = 0xFF00; // set portC to output
    TRISE = 0xFF00; // set portE to output

    TRISD = 0xFFFF; // set portD to input

    while (1)
    {
        for (i = 1001L * 100; i > 0; i--)Nop(); // delay start with 0

        playerCount = 1;
        currentTurn = 0;

        // records player count
        while (PORTDbits.RD13 != 0)
        {
            if (PORTDbits.RD6 == 0) // PORTAbits.RA0 == 1
            {
                playerCount = 2;
            }
            else if(PORTDbits.RD7 == 0) //PORTAbits.RA1 == 1
            {
                playerCount = 3;
            }
            else if(PORTAbits.RA7 == 0) //PORTAbits.RA2 == 1
            {
                playerCount = 4;
            }
        }

        // starts game
        maxTurn = startGame (playerCount, maxTurn);
        ms_delay (500);

        // runs until all players have played
        while(currentTurn < maxTurn)
```

```
        {
            while(turnFlag)
            {
                if(PORTDbits.RD6 == 0)
                {
                    hit ();
                }
                else if(PORTDbits.RD7 == 0)
                {
                    turnFlag = 0;
                }
                else if(PORTAbits.RA7 == 0)
                {
                    turnFlag = 0;
                }
            }
            turnFlag = 1;
            currentTurn = currentTurn + 1;

            if(currentTurn < maxTurn - 1)
            {
                rotate ();
            }
            else if(currentTurn == maxTurn - 1)
            {
                if(playerCount == 1)
                {
                    break;
                }
                else
                {
                    if(playerCount == 2)
                    {
                        inverseRotate();
                    }
                    else if(playerCount == 3)
                    {
                        inverseRotate();
                        inverseRotate();
                    }
                    else if (playerCount == 4)
                    {
                        inverseRotate();
                        inverseRotate();
                        inverseRotate();
                    }
                }
            }
```

```
            ms_delay (1000);
        }
    }
    return 1;
}
```

(DD, JR)

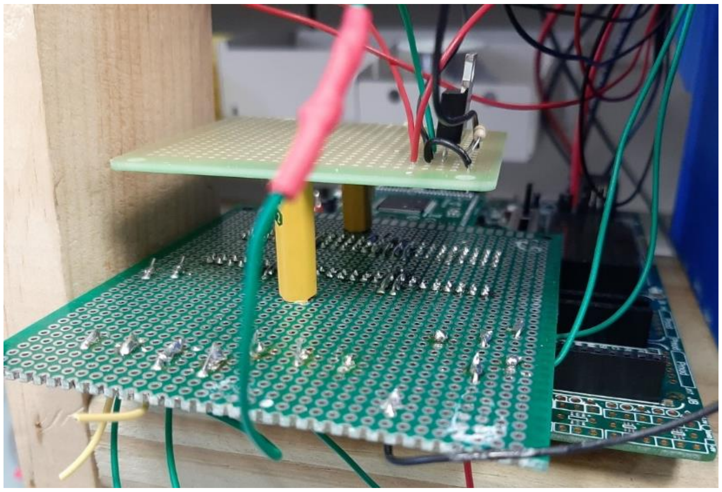


Figure 18: Explorer16/32 Daughter Board

This picture shows the daughter board that was used to interface with the Explorer16/32 development board. This was used to plug into the dev board and was used to re-route the different ports that were being used in the code. This was done to prevent soldering directly onto the dev board and to allow more space for stable wire connections. The board mounted on top of the daughter board is a driver circuit. (DD)

6. Mechanical Sketch

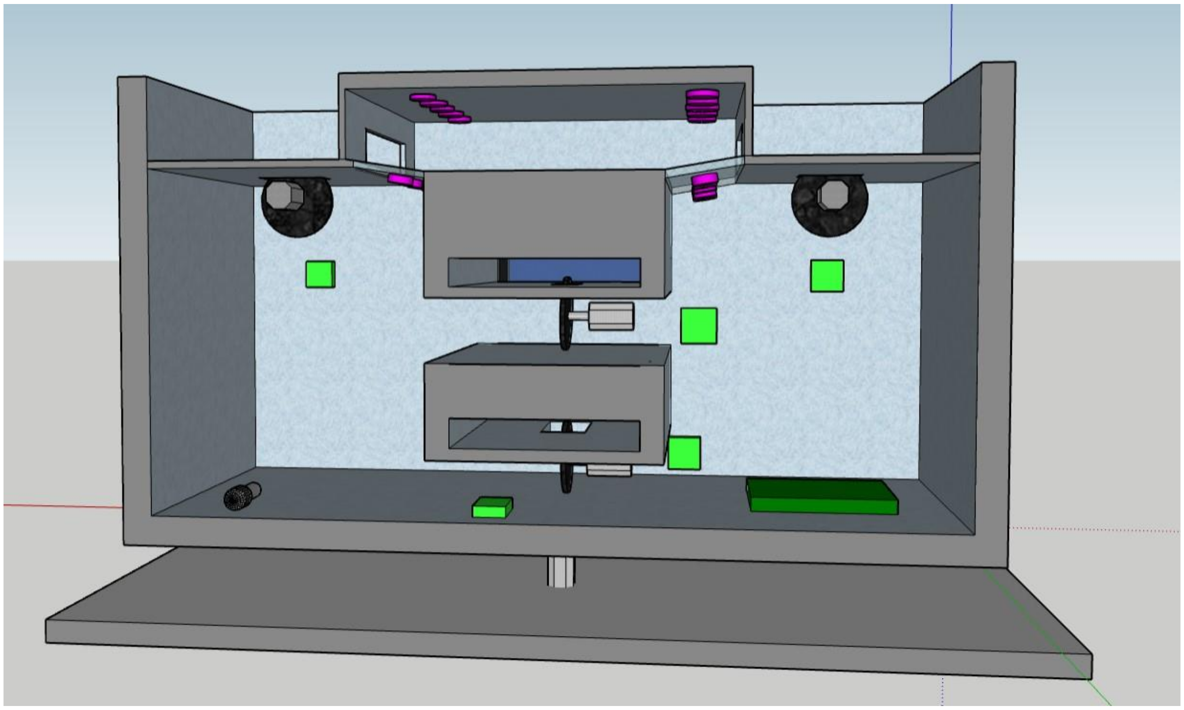


Figure 19: Mechanical Sketch Front View

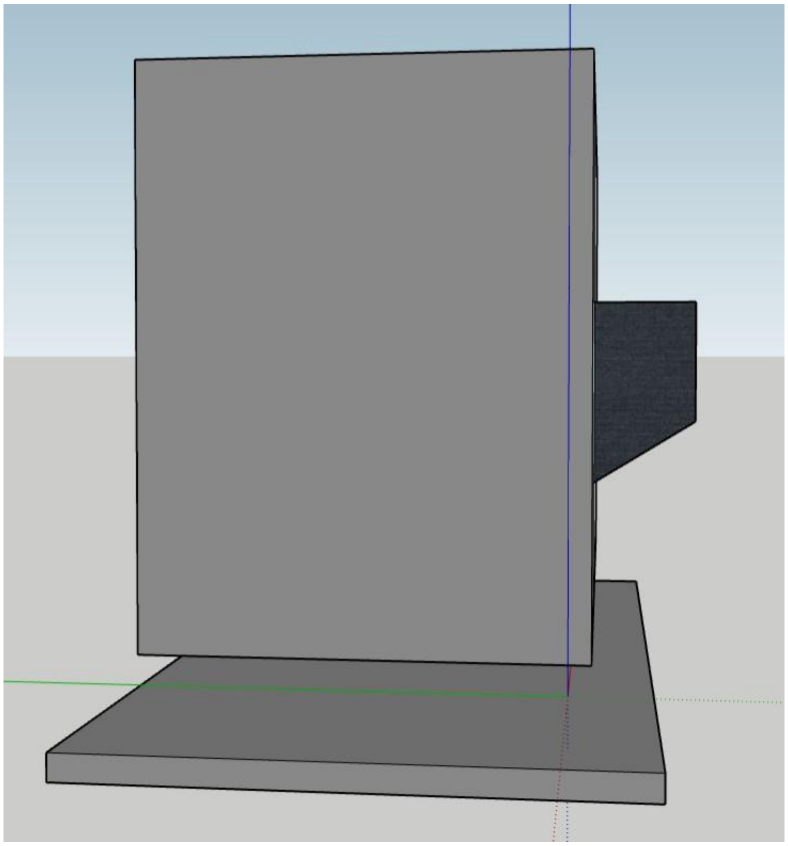


Figure 20: Mechanical Sketch Side View

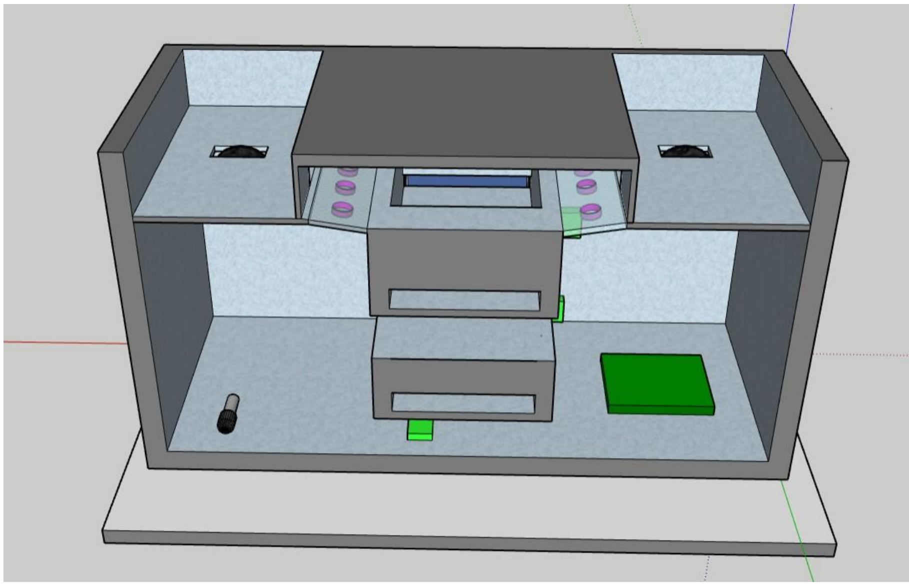


Figure 21: Mechanical Sketch Overhead View

This is the mechanical sketch for our current design of the card dealing robot and most everything can be seen in Figure 19. The dark green block represents the Explorer 32 dev kit board that we will be using, and the five bright green blocks represent the five individual motor driver circuits. The motors in our sketch are shown as gray cylinders where four of them will be moving cards and one of them is at the bottom of our robot which will rotate the entire body of the design. The purple cylinders represent the UV-C LED's that will be used for sanitization. There is a microphone in the bottom left of our design that will be used to access the voice input for the speech recognition. Due to our change of scope, the microphone will not be in use. Instead, we will be replacing the microphone with four external and wired keypads that will be used for taking in the player commands. There will be a card chute (represented as a black box in Figure 20) that will be on the back of our design that will have the purpose of flipping select cards face up. (DD)

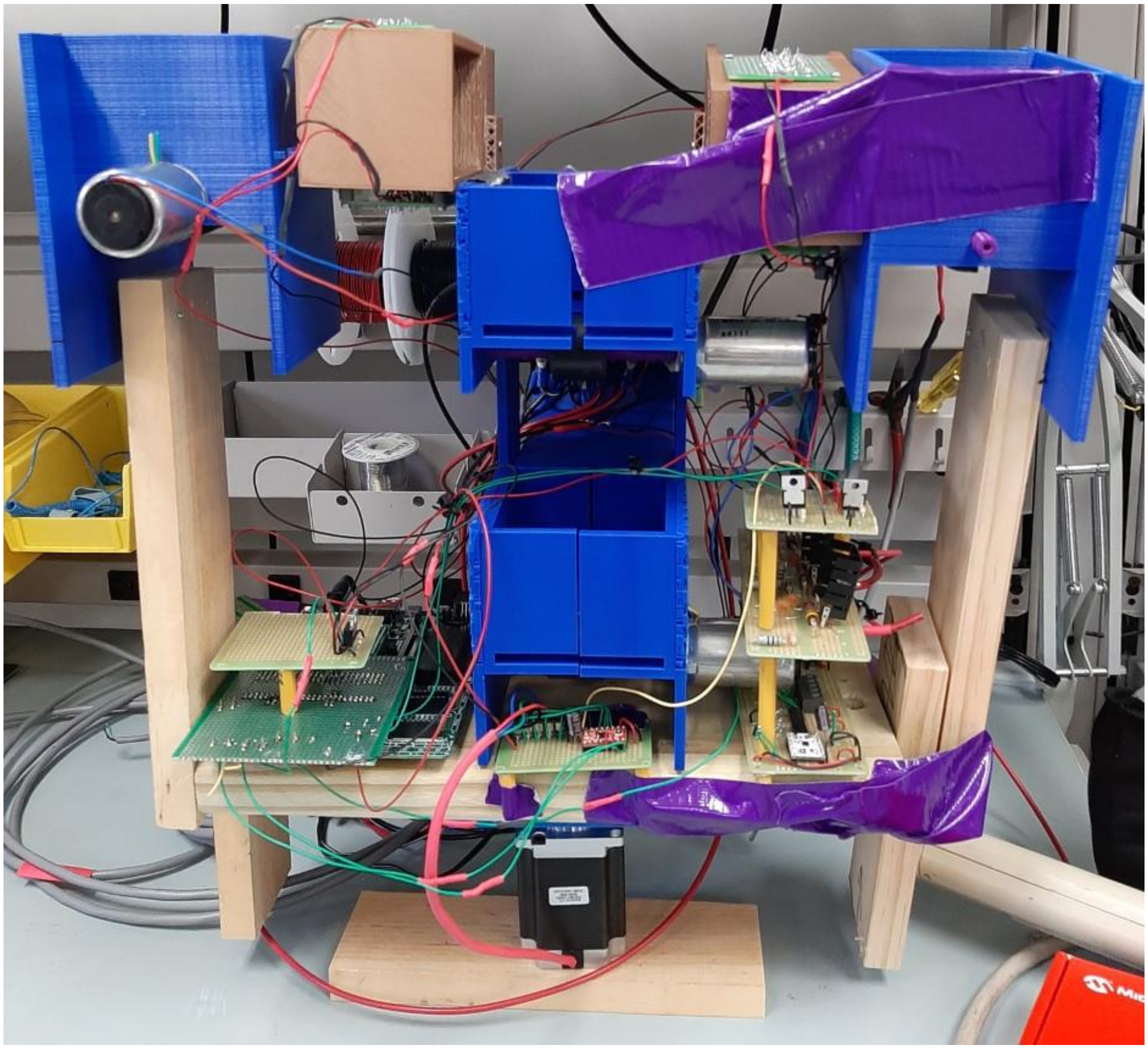


Figure 22: The Final Project Front View

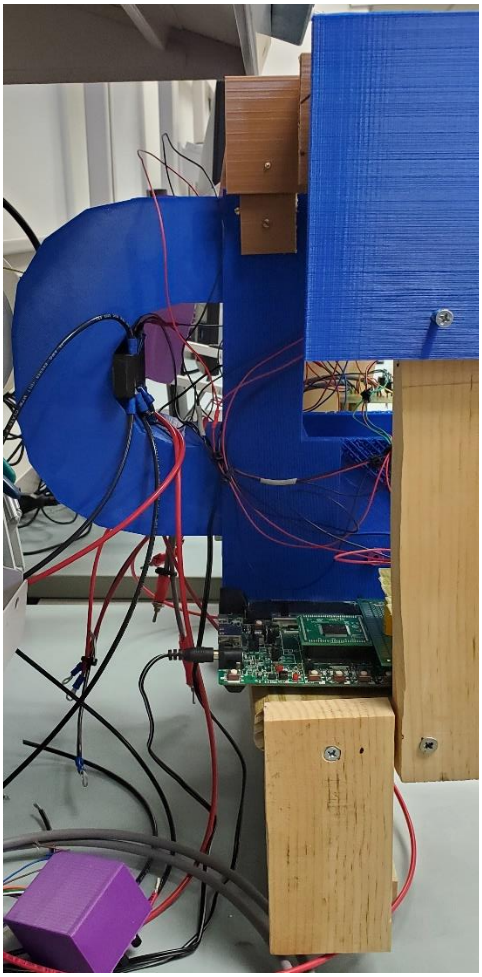


Figure 23: The Final Project Side View

This is the final prototype of the project. Wood was used to mount the project since the external housing could not properly be 3D printed. This resulted in some mechanical issues. (DD)

7. Design Team Information

- Grant Crump, Electrical Engineering. ESI (N).
- Drake Drivere, Computer Engineering. ESI (Y).
- Adam Miner, Electrical Engineering. ESI (N).
- Jakob Reiter, Computer Engineering. ESI (Y).

8. Parts Lists

Parts List

Qty.	Refdes	Part Num.	Description
16	SW1 thru SW4	PS1024ABLK	SWITCH PUSH SPST-NO 3A 125V
4	Rd	EP3WS100RJ	RES 100 OHM 5% 3W AXIAL
16	Rhit/stay/bust/active	CFM14JT200R	RES 200 OHM 5% 1/4W AXIAL
7	U1	MCP6141T-E/OT	IC OPAMP GP 1 CIRCUIT SOT23-5
21	Rf/R2	RNF18FTD100K	RES 100K OHM 1% 1/8W AXIAL
4	MOTORS	PF25-48Q4	TIN-CAN STEP MOTOR 48 STEPS 5V
3	DRIVERS	2876	STSPIN220 LOW-VOLT STPR MTR DRVR
1	CART	CART	SHOPPING CART CHECKOUT - ADAM MINER
1	Voltage Reg circuit R1, R	EFR-W0P25-A:MF	EDGELEC 100pcs 240 ohm Resistor 1/4w (0.25 Watt) ±1% Tolerance Metal Film Fixed Resistor
8	Voltage Reg C2-C9	FG24X7R1A106KRT00	CAP CER 10UF 10V X7R RADIAL
1	Voltage Reg C1	1C25Z5U104M050B	CAPACITOR, 0.1UFD, 50VDC, RADIAL
8	D1-D8	1N4148	DIODE,DO-35,100V,100MA,4NS,FAST.
4	LT1085	LT1085IT-12#PBF	LDO Voltage Regulators 12V Low Dropout Pos VR 3A
1	UVC-Circuit D1-10	100F5W-YT-WH-WH	Chanzon 100 pcs 5mm White Diffused LED Diode Lights (Frosted Lens Round DC 3V 20mA)

(AM, GC)

Budget

Qty.	Part Num.	Description	Unit Cost	Total Cost
16	PS1024ABLK	SWITCH PUSH SPST-NO 3A 125V	$1.63	$26.08
4	EP3WS100RJ	RES 100 OHM 5% 3W AXIAL	0.54	2.16
16	CFM14JT200R	RES 200 OHM 5% 1/4W AXIAL	0.10	1.60
7	MCP6141T-E/OT	IC OPAMP GP 1 CIRCUIT SOT23-5	0.67	4.69
21	RNF18FTD100K	RES 100K OHM 1% 1/8W AXIAL	0.10	2.10
4	PF25-48Q4	TIN-CAN STEP MOTOR 48 STEPS 5V	26.00	104.00
3	2876	STSPIN220 LOW-VOLT STPR MTR DRVR	5.95	17.85
1	CART	SHOPPING CART CHECKOUT - ADAM MINER		
1	EFR-W0P25-A:MI	EDGELEC 100pcs 240 ohm Resistor 1/4w (0.25 Watt) ±1% T	4.99	4.99
8	FG24X7R1A106KI	CAP CER 10UF 10V X7R RADIAL	0.56	4.48
1	1C25Z5U104M05(	CAPACITOR, 0.1UFD, 50VDC, RADIAL	0.20	0.20
8	1N4148	DIODE,DO-35,100V,100MA,4NS,FAST.	0.10	0.80
4	LT1085IT-12#PBF	LDO Voltage Regulators 12V Low Dropout Pos VR 3A	12.55	50.20
1	100F5W-YT-WH-\	Chanzon 100 pcs 5mm White Diffused LED Diode Lights (Fro	5.88	5.88
			Total	$225.03

(AM, GC)

Qty.	Refdes	Part Num.	Description	Suggested Vendor	Vendor Part Num.	Catalog #/Page #/Hyperlink	Unit Cost	Total Cost
4		LM338T/NOPB	Linear Voltage Regulators 5 AMP ADJUSTABLE REG	Mouser	926-LM338T/NOPB	https://www.mouser.com/Produ	$2.45	$9.80
1		1370N15	Multipurpose Neoprene Rubber Sheet with Certificat	McMaster-CARR	1370N15	mcmaster.com/rubber/rubber/m	10.14	10.14
							Total	$19.94

The only addition to the budget was $19.94 this semester. This brings the grand total for the project to $244.97. This beats our budget goal of $600. (AM)

9. Project Schedules

Preliminary Schedule

Task Name	Duration	Start	Finish	Predecessors	Resource Names
SDP I 2021					Adam Miner,Drake Driv
Project Design	**191.38 days**	**Wed 8/25/21**	**Fri 3/4/22**		**Adam Miner,Drake Dr**
Midterm Report	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Cover page	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
T of C, L of T, L of F	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Problem Statement	**44.38 days**	**Wed 8/25/21**	**Fri 10/8/21**		**Adam Miner,Drake Dr**
Need	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Objective	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Background	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Marketing Requirements	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Engineering Requirements Specification	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Engineering Analysis	**44.38 days**	**Wed 8/25/21**	**Fri 10/8/21**		**Adam Miner,Drake Dr**
Circuits (DC, AC, Power, ...)	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Electronics (analog and digital)	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Signal Processing	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Communications (analog and digital)	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Electromechanics	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Computer Networks	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Embedded Systems	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Controls	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Accepted Technical Design	**44.38 days**	**Wed 8/25/21**	**Fri 10/8/21**		**Adam Miner,Drake Dr**
Hardware Design: Phase 1	**44.38 days**	**Wed 8/25/21**	**Fri 10/8/21**		**Adam Miner,Drake Dr**
Software Design: Phase 1	**44.38 days**	**Wed 8/25/21**	**Fri 10/8/21**		**Adam Miner,Drake Dr**
Software Behavior Models Levels 0 thru N (w/FR tables)	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Mechanical Sketch	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Team information	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Project Schedules	**44.38 days**	**Wed 8/25/21**	**Fri 10/8/21**		**Adam Miner,Drake Dr**
Midterm Design Gantt Chart	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
References	46 days	Wed 8/25/21	Fri 3/4/22		Adam Miner,Drake Driv
Midterm Parts Request Form	49.38 days	Wed 8/25/21	Wed 10/13/21		Adam Miner,Drake Driv
Midterm Design Presentations Day 1	0 days	Wed 9/22/21	Wed 9/22/21		Adam Miner,Drake Driv
Midterm Design Presentations Day 2	0 days	Wed 9/29/21	Wed 9/29/21		Adam Miner,Drake Driv
Project Poster	12.38 days	Sun 10/10/21	Fri 10/22/21		Adam Miner,Drake Driv
Final Design Report	44.38 days	Sun 10/10/21	Tue 11/23/21	3	Adam Miner,Drake Driv
Abstract	44.38 days	Sun 10/10/21	Tue 11/23/21	3	Adam Miner,Drake Driv
Hardware Design: Phase 2	**44.38 days?**	**Sun 10/10/21**	**Tue 11/23/21**	**3**	**Adam Miner,Drake Dr**
Modules 1...n	**44.38 days**	**Sun 10/10/21**	**Tue 11/23/21**	**3**	**Adam Miner,Drake Dr**
Simulations	44.38 days	Sun 10/10/21	Tue 11/23/21	3	Adam Miner,Drake Driv
Schematics	44.38 days	Sun 10/10/21	Tue 11/23/21	3	Adam Miner,Drake Driv
Software Design: Phase 2	**44.38 days**	**Sun 10/10/21**	**Tue 11/23/21**		**Adam Miner,Drake Dr**
Modules 1...n	**44.38 days**	**Sun 10/10/21**	**Tue 11/23/21**		**Adam Miner,Drake Dr**
Code (working subsystems)	44.38 days	Sun 10/10/21	Tue 11/23/21	3	Adam Miner,Drake Driv
System integration Behavior Models	44.38 days	Sun 10/10/21	Tue 11/23/21	3	Adam Miner,Drake Driv
Parts Lists	**44.38 days**	**Sun 10/10/21**	**Tue 11/23/21**		**Adam Miner,Drake Dr**
Parts list(s) for Schematics	44.38 days	Sun 10/10/21	Tue 11/23/21	3	Adam Miner,Drake Driv
Materials Budget list	44.38 days	Sun 10/10/21	Tue 11/23/21	3	Adam Miner,Drake Driv
Proposed Implementation Gantt Chart	44.38 days	Sun 10/10/21	Tue 11/23/21	3	Adam Miner,Drake Driv
Conclusions and Recommendations	44.38 days	Sun 10/10/21	Tue 11/23/21	3	Adam Miner,Drake Driv
Parts Request Form for Subsystems	35.38 days	Wed 9/22/21	Wed 10/27/21	32	Adam Miner,Drake Driv
Subsystems Demonstrations Day 1	0 days	Wed 11/10/21	Wed 11/10/21		Adam Miner,Drake Driv
Subsystems Demonstrations Day 2	0 days	Wed 11/17/21	Wed 11/17/21		Adam Miner,Drake Driv
Parts Request Form for Spring Semester	8.38 days	Tue 11/23/21	Wed 12/1/21	35	Adam Miner,Drake Driv

(AM)

Senior Design Proposal

Implemented Schedule

		Task Mode	Task Name	Duration	Start	Finish		Resource Names
1			**SDP2 Implementation 2022**	**453 days?**	**Mon 1/10/22**	**Fri 4/7/23**		
2			Revise Gantt Chart	14 days	**Mon 1/10/22**	Sun 1/23/22		
3			**Implement Project Design**	**89 days?**	**Mon 1/10/22**	**Fri 4/8/22**		
4			**Hardware Implementation**	**42 days?**	**Mon 1/10/22**	**Sun 2/20/22**		
5			**Breadboard Components**	**14 days?**	**Mon 1/10/22**	**Sun 1/23/22**		
6			Shuffling	14 days	Mon 1/10/22	Sun 1/23/22		Adam Miner
7			Dealing	14 days	Mon 1/10/22	Sun 1/23/22		Adam Miner
8			Voltage Regulation	14 days	Mon 1/10/22	Sun 1/23/22		Grant Crump
9			UV-C LEDs	14 days	Mon 1/10/22	Sun 1/23/22		Grant Crump
10			Keypads	14 days	Mon 1/10/22	Sun 1/23/22		Adam Miner
11			**Layout and Generate Solderboard(s)**	**14 days?**	**Mon 1/10/22**	**Sun 1/23/22**		
12			Shuffling	14 days	Mon 1/10/22	Sun 1/23/22		Adam Miner
13			Dealing	14 days	Mon 1/10/22	Sun 1/23/22		Adam Miner
14			Voltage Regulation	14 days	Mon 1/10/22	Sun 1/23/22		Grant Crump
15			UV-C LEDs	14 days	Mon 1/10/22	Sun 1/23/22		Grant Crump
16			Keypads	14 days	Mon 1/10/22	Sun 1/23/22		Adam Miner
17			**Assemble Hardware**	**14 days**	**Mon 1/24/22**	**Sun 2/6/22**	**16**	
18			Shuffling	14 days	Mon 1/24/22	Sun 2/6/22	16	Adam Miner
19			Dealing	14 days	Mon 1/24/22	Sun 2/6/22	16	Adam Miner
20			Voltage Regulation	14 days	Mon 1/24/22	Sun 2/6/22	16	Grant Crump
21			UV-C LEDs	14 days	Mon 1/24/22	Sun 2/6/22	16	Grant Crump
22			Keypads	14 days	Mon 1/24/22	Sun 2/6/22	16	Adam Miner
23			**Test Hardware**	**7 days**	**Mon 2/7/22**	**Sun 2/13/22**	**22**	
24			Shuffling	7 days	Mon 2/7/22	Sun 2/13/22	22	Adam Miner
25			Dealing	7 days	Mon 2/7/22	Sun 2/13/22	22	Adam Miner

		Task Mode	Task Name	Duration	Start	Finish		Resource Names	Add New Column
23			**Test Hardware**	**7 days**	**Mon 2/7/22**	**Sun 2/13/22**	**22**		
24			Shuffling	7 days	Mon 2/7/22	Sun 2/13/22	22	Adam Miner	
25			Dealing	7 days	Mon 2/7/22	Sun 2/13/22	22	Adam Miner	
26			Voltage Regulation	7 days	Mon 2/7/22	Sun 2/13/22	22	Grant Crump	
27			UV-C LEDs	7 days	Mon 2/7/22	Sun 2/13/22	22	Grant Crump	
28			Keypads	7 days	Mon 2/7/22	Sun 2/13/22	22	Adam Miner	
29			**Revise Hardware**	**7 days**	**Mon 2/14/22**	**Sun 2/20/22**	**28**		
30			Shuffling	7 days	Mon 2/14/22	Sun 2/20/22	28	Adam Miner	
31			Dealing	7 days	Mon 2/14/22	Sun 2/20/22	28	Adam Miner	
32			Voltage Regulation	7 days	Mon 2/14/22	Sun 2/20/22	28	Grant Crump	
33			UV-C LEDs	7 days	Mon 2/14/22	Sun 2/20/22	28	Grant Crump	
34			Keypads	7 days	Mon 2/14/22	Sun 2/20/22	28	Adam Miner	
35			*MIDTERM: Demonstrate Hardware Subsystems*	5 days	Mon 2/21/22	Fri 2/25/22	29		
36			SDC & FA Hardware Approval	0 days	Sat 2/26/22	Sat 2/26/22	29		
37			**Software Implementation**	**42 days**	**Mon 1/10/22**	**Sun 2/20/22**			
38			**Develop Software**	**28 days**	**Mon 1/10/22**	**Sun 2/6/22**			
39			Keypads	28 days	Mon 1/10/22	Sun 2/6/22		Jakob Reiter	
40			Intercommunications	28 days	Mon 1/10/22	Sun 2/6/22		Drake Drivere	
41			**Test Software**	**28 days**	**Mon 1/10/22**	**Sun 2/6/22**			
42			Keypads	28 days	Mon 1/10/22	Sun 2/6/22		Jakob Reiter	
43			Intercommunications	28 days	Mon 1/10/22	Sun 2/6/22		Drake Drivere	
44			**Revise Software**	**14 days**	**Mon 2/7/22**	**Sun 2/20/22**	**38**		
45			Keypads	14 days	Mon 2/7/22	Sun 2/20/22	38	Jakob Reiter	
46			Intercommunications	14 days	Mon 2/7/22	Sun 2/20/22	38	Drake Drivere	
47			*MIDTERM: Demonstrate Software Subsystems*	5 days	Mon 1/10/22	Fri 1/14/22			

		Task Mode	Task Name	Duration	Start	Finish		Resource Names
46			Intercommunications	14 days	Mon 2/7/22	Sun 2/20/22	38	Drake Drivere
47			*MIDTERM: Demonstrate Software Subsystems*	5 days	Mon 1/10/22	Fri 1/14/22		
48			SDC & FA Software Approval	0 days	Mon 2/21/22	Mon 2/21/22		
49			**System Integration**	**91 days**	**Sat 1/15/22**	**Fri 4/15/22**		
50			**Assemble Complete System Integration**	**14 days**	**Sat 1/15/22**	**Fri 1/28/22**	**47**	
51			<subsystem 1>	14 days	Sat 1/15/22	Fri 1/28/22	47	Adam Miner
52			<subsystem 2>	14 days	Sat 1/15/22	Fri 1/28/22	47	Grant Crump
53			<subsystem 3>	14 days	Sat 1/15/22	Fri 1/28/22	47	Jakob Reiter
54			<subsystem n>	14 days	Sat 1/15/22	Fri 1/28/22	47	Drake Drivere
55			**Test Complete System Integration**	**7 days**	**Sat 1/29/22**	**Fri 2/4/22**	**50**	
56			<subsystem 1>	7 days	Sat 1/29/22	Fri 2/4/22	50	Adam Miner
57			<subsystem 2>	7 days	Sat 1/29/22	Fri 2/4/22	50	Grant Crump
58			<subsystem 3>	7 days	Sat 1/29/22	Fri 2/4/22	50	Jakob Reiter
59			<subsystem n>	7 days	Sat 1/29/22	Fri 2/4/22	50	Drake Drivere
60			**Revise Complete System Integration**	**24 days**	**Sat 3/19/22**	**Mon 4/11/22**	**55**	
61			<subsystem 1>	24 days	Sat 3/19/22	Mon 4/11/22	55	Adam Miner
62			<subsystem 2>	24 days	Sat 3/19/22	Mon 4/11/22	55	Grant Crump
63			<subsystem 3>	24 days	Sat 3/19/22	Mon 4/11/22	55	Jakob Reiter
64			<subsystem n>	24 days	Sat 3/19/22	Mon 4/11/22	55	Drake Drivere
65			*Preliminary Demonstration of Complete System*	4 days	Tue 4/12/22	Fri 4/15/22	60	
66			**Develop Final Report**	**103 days**	**Mon 1/10/22**	**Fri 4/22/22**		
67			Write Final Report	103 days	Mon 1/10/22	Fri 4/22/22		
68			Submit Final Report	0 days	Fri 4/22/22	Fri 4/22/22	67	
69			Spring Recess	7 days	Mon 3/21/22	Sun 3/27/22		
70			*Project Demonstration and Presentation*	0 days	Mon 4/18/22	Mon 4/18/22		

The above Gantt charts were effective for project management. The team kept to the schedule by and large. (AM)

10. Conclusions and Recommendations

This team has done a significant amount of research on the different subject areas that pertain to this project. The team tested different design ideas, techniques, performed simulations as well as prototyping. The motor operation and microprocessor control have been tested. The plan to implement voice recognition has been scrapped due to time constraints and is going to be replaced with keypad controllers for player commands. The remaining focus of the team will be directed towards system integration to get a fully operational prototype. The final implementation includes motors to deal, motors to shuffle, a motor to rotate the entire structure. External keypads were attempted in the final implementation but were unsuccessful. Instead, the on-board buttons of the Explorer16/32 were used to control the robot. This did not change much with the code but removed the need for the button circuit. Most of the issues with the implementation involved mechanical issues. These included friction on the axels, which could've been rectified by using bearings. Another mechanical issue was the structure was not able to be printed, so some of the paths for the cards to go did not line up. The team dynamic was satisfactory throughout the entire project, the team worked together to work through all issues, and everyone pulled their weight. The team has no plans to continue the development of this project. Recommendations for future students include eliminating as many mechanical components as possible unless they know how to tackle said components. (DD, JR)

11. References

[1] "How Much Do Casino Dealers Make? | Gamblers Daily Digest," Gamblers Daily Digest -, Feb. 02, 2021.

[2] F. Jelinek, Statistical Methods for Speech Recognition. MIT Press, 1997.

[3] E. I. Agustin, R. T. Yunardi, and A. Firdaus, "Voice recognition system for controlling electrical appliances in smart hospital room" Uakron.edu, 2021. http://web.a.ebscohost.com.ezproxy.uakron.edu:2048/ehost/pdfviewer/pdfviewer?vid=34&sid=01e2f22b-c46f-4d89-b783-c2978d05a3c2%40sessionmgr4007 (accessed Mar. 20, 2021).

[4] "System for audio distribution including network microphones for voice applications." https://patents.google.com/patent/US20190103113A1/en?q=voice+recognition+circuit&oq=voice+recognition+circuit (accessed Mar. 20, 2021).

[5] Lai and Nunayon, "LEDs require no warm-up time, less energy to operate, and exhibit a very long operational lifespan up to 100,000 hours," https://onlinelibrary-wiley-com.ezproxy.uakron.edu:2443/doi/full/10.1111/ina.12752.

[6] Mackenzie, Dana. "Ultraviolet Light Fights New Virus." Engineering (Beijing, China), 27 June 2020, www.ncbi.nlm.nih.gov/pmc/articles/PMC7319933/, 10.1016/j.eng.2020.06.009.

[7] Arvid and Daruish, "Spelkortsdelaren The Card Dealer." Accessed: Mar. 20, 2021. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1462013/FULLTEXT01.pdf.

[8] Daiconis, et al. "Analysis of Casino Shelf Shuffling Machines." , projecteuclid.org/journals/annals-of-applied-probability/volume-23/issue-4/Analysis-of-casino-shelf-shuffling-machines/10.1214/12-AAP884.full?tab=ArticleFirstPage.

[9] Grauzer, Attila, et al. Playing Card Delivery for Games with Multiple Dealing Rounds. patents.google.com/patent/AU2007254321B2/en?q=mechanical+playing+card+shuffler&oq=mechanical+playing+card+shuffler. Accessed 20 Mar. 2021.

[10] Lu, B. (2020). *Mean Well Enterprises Co., ltd.*. UV-C STERILIZATION LED LIGHT and LED Drivers -MEAN WELL Switching Power Supply Manufacturer. Retrieved October 2, 2021, from https://www.meanwell.com/newsInfo.aspx?c=5&i=906.

(GC, DD, AM, JR)

12. Appendices

1. STSPIN220 Datasheet
2. LT1085 Datasheet
3. PF25-48Q4 Datasheet
4. SY57STH76-2804A Datasheet