Individual Progress Report
Autonomous Card Shuffler & Dealer
ECE 445
Spring 2023

Adam Naboulsi

Table Of Contents

1. Introduction

The card dealer and shuffler that my team has been working on aims to ensure fairness in the shuffling and dealing process when it comes to card games whether played in a social setting or at a casino. Popular examples include Poker, Blackjack, and Kings Corner. The card shuffler eliminates human error and speeds up the dealing process as the whole system can rotate and deal cards in an efficient fashion. Our project is split up into four main subsystems: Power, User Interface, Sensing, and Motor. These subsystems will all work together and communicate with one another through the centralized microcontroller.

When it came to the design process – mechanical design, schematic design, and PCB design – we all contributed equally as a team. In terms of testing and programming, however, my focus has been on the motor subsystem. We have completed the mechanical design and have retrieved our mechanical system; we have completed our schematic design and PCB design, and have sent in our orders for our first round of PCBs. I performed some testing on the dealing and shuffling motors of our motor subsystem to ensure proper functionality for when we solder our PCB, mount it onto the mechanical system, and start putting everything together. I am currently working on programming the microcontroller pins designated to the motor subsystem. Once I finish programming the microcontroller pins corresponding to the motor subsystem and my partners program the other pins for the other subsystems, we will then start attaching all the components to our mechanical system and test the unit as a whole.

2. Design

2.1 Design Procedure:

When it came to designing our project we first began with creating a block diagram and sketching the mechanical design. We divided our project into the four main subsystems discussed earlier and discussed how each subsystem would interact with the others through a microcontroller chip. We then created a schematic, had it reviewed, and finally designed our pcb. In regard to the mechanical design we initially created, we reviewed it with the mechanical team, discussed necessary adjustments, and finalized the design.

2.2 Design Details:

2.2.1 Electrical Design:

Our final design consisted of the following four subsystems with the STM32F103C8T6 microcontroller chip in the center: Power, Motor, Sensing, and User Interface. The power subsystem's purpose is to act as a stable and reliable power source for the various subsystems and their corresponding components. The motor subsystem's purpose is to control both the dealing and shuffling processes in the project – my main focus within the team. The purpose of the sensing subsystem is to provide us with distance readings as the device spins around in such a way that the device can know when a player is facing it and thus deal the cards in an organized manner. Lastly, the purpose of the User Interface subsystem is to provide the user with the ability to select the current card game and the current number of players, and control when to shuffle or deal through push buttons and a display. These components will all work together to allow the system to shuffle and deal the cards efficiently around a table. A diagram below illustrates the basic communication system we designed among the subsystems:

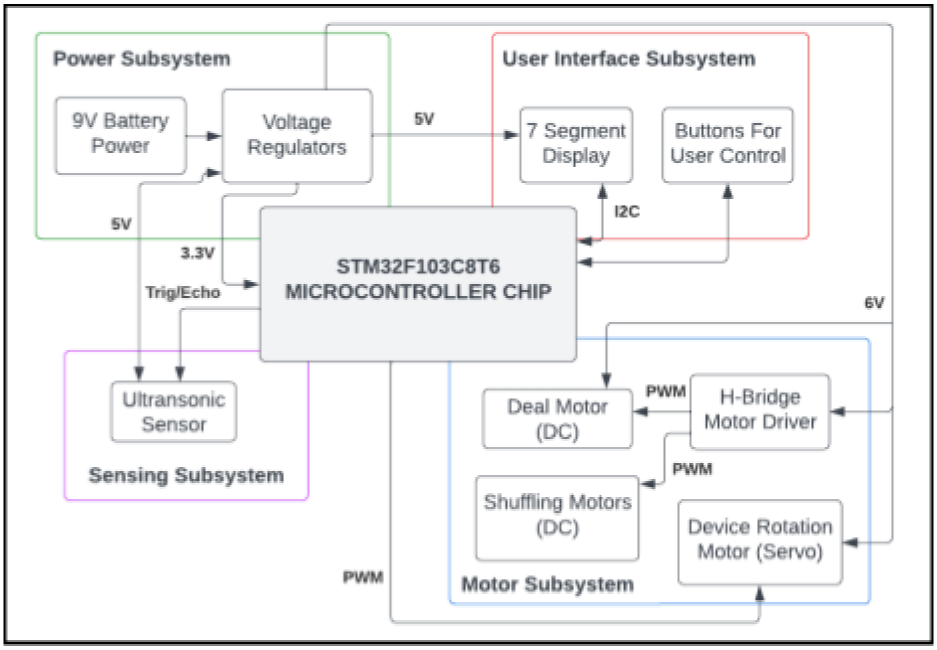

Fig 2.1 *Block Diagram*

2.2.2 Mechanical Design:

In regards to the mechanical design, we finalized our design with the mechanical team to ensure proper functionality. Below is a series of figures showing the mechanical design from a top and bottom view. Our design involves a base holding the card shuffling and dealing system above it in place with a servo motor that can rotate the system as a whole 180 degrees. The card shuffler section consists of two dc motors to shuffle the cards individually into a card holder slot, and an ultrasonic sensor mounted on the outside to determine the distance to whatever obstacle it is facing. Once the cards are shuffled and held in the middle slot, our dealing section of the system will function as the base rotates the system around a table; it consists of a dc motor beneath the card slot with a lever that pokes out under the bottom of the card dealer such that it can grip onto the cards. On the outside of our system we designed the 7-segment display along with the buttons for parameter adjustment to be mounted on the outside of the enclosure.

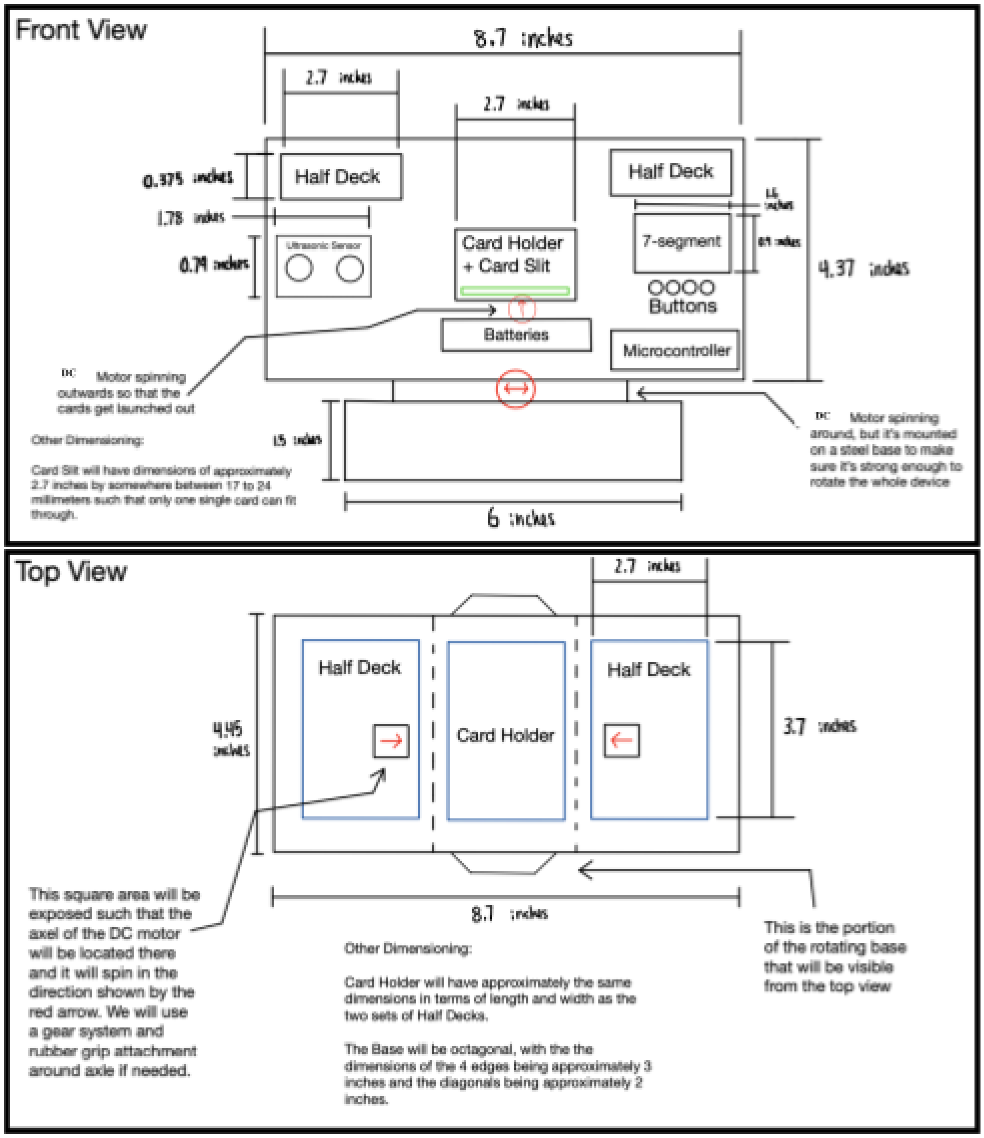

Fig 2.2 *Mechanical Design (Top & Front View)*

2.2.3 Motor Subsystem Schematic Design:

As mentioned earlier, my main focus among our team for testing and some of the design process was the motor subsystem. Firstly, as context, below is a schematic of our full system along with a zoomed in image of the motor subsystem and the centralized microcontroller unit.

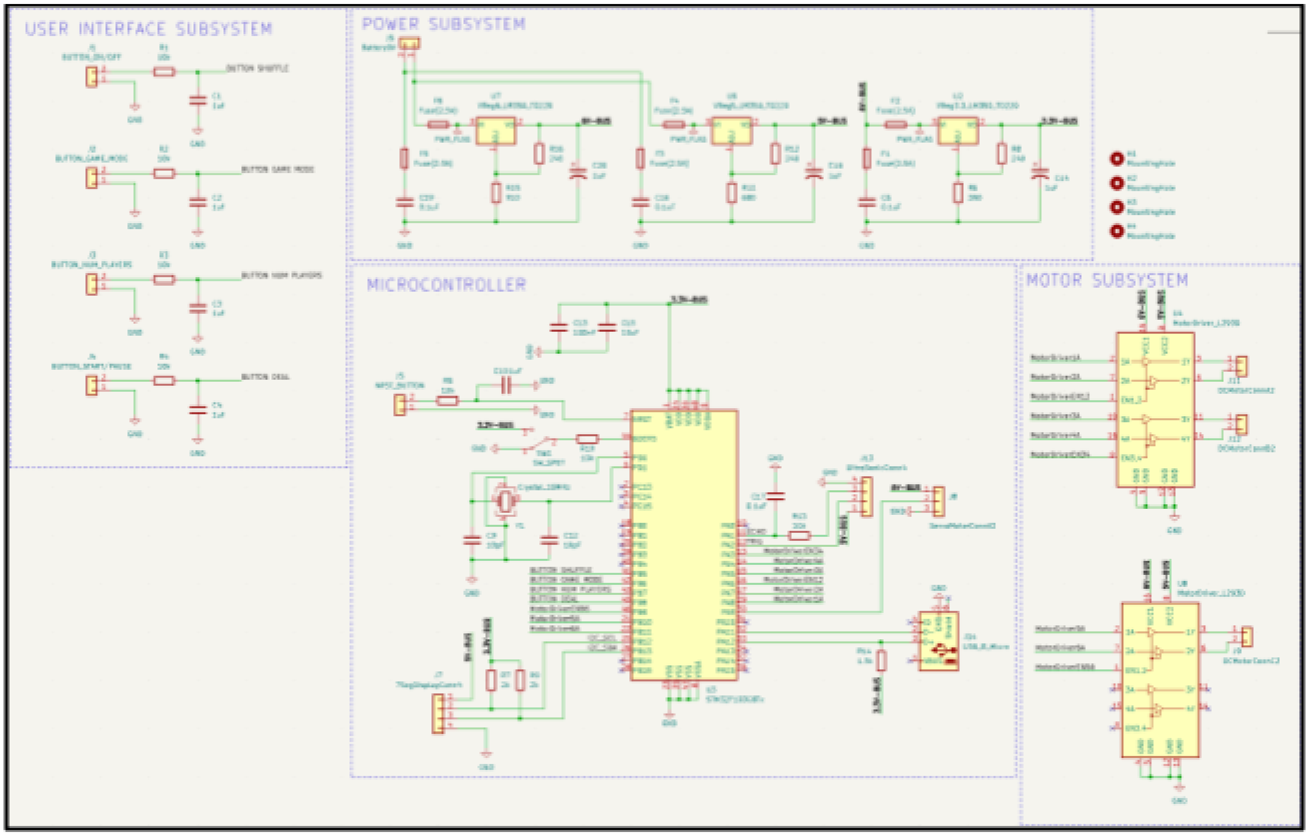

Fig 2.3 *Full Schematic Design*

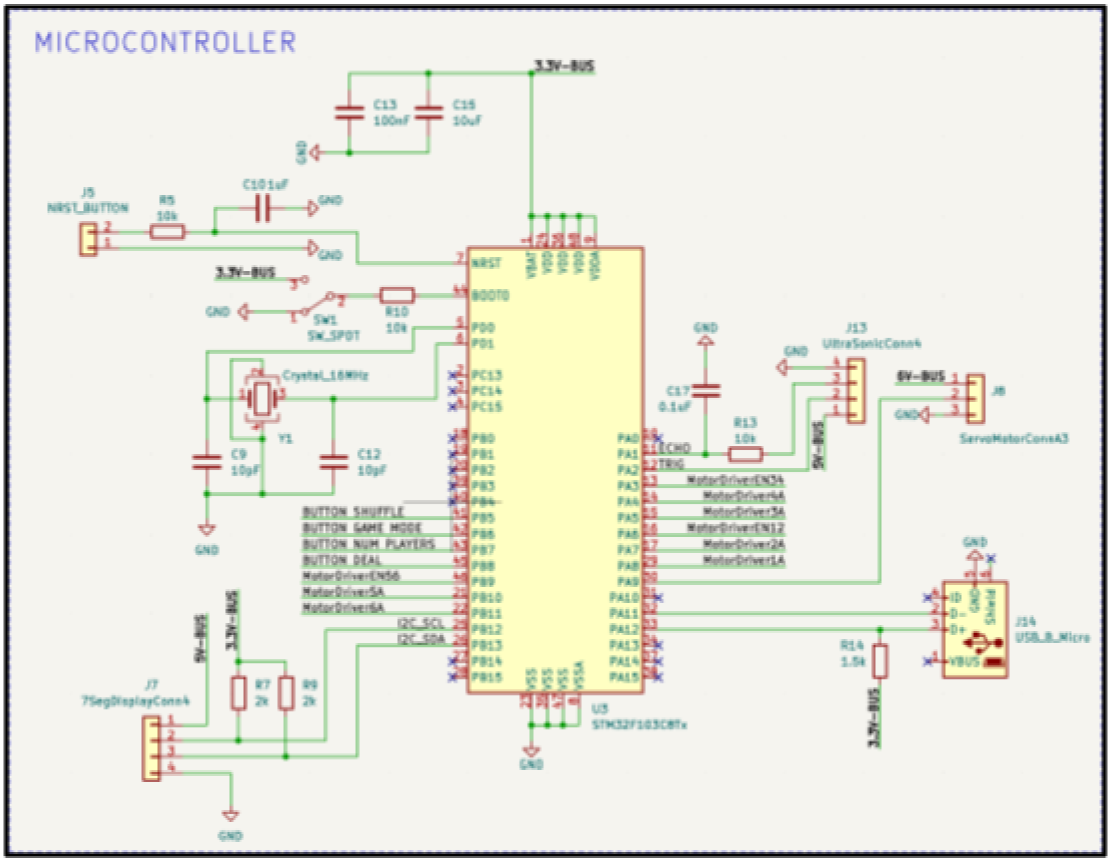

Fig 2.4 *Microcontroller Schematic Design*

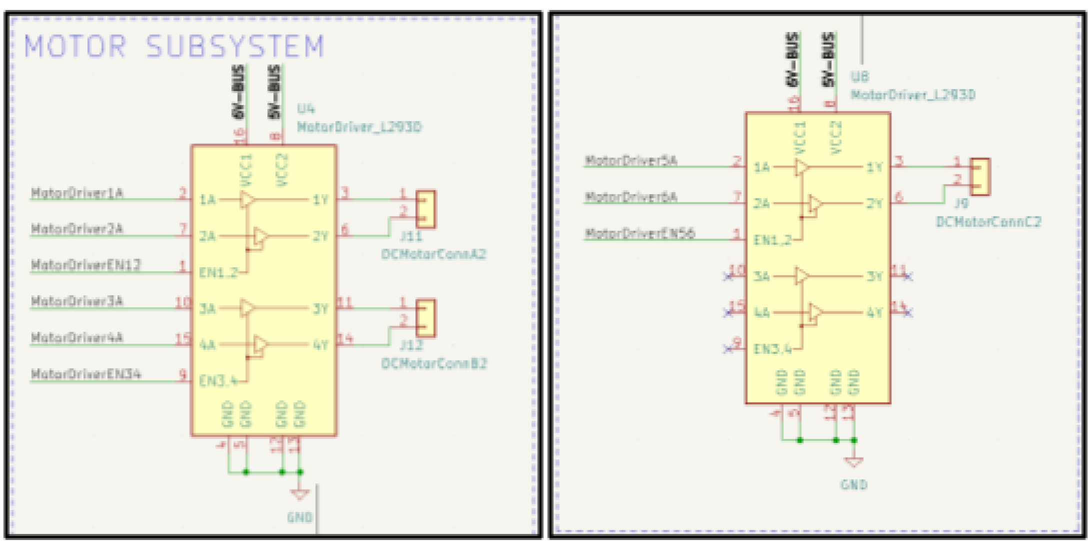

Fig 2.5 *Motor Subsystem Schematic Design*

As mentioned earlier, each subsystem interfaces intimately with the microcontroller. The motor subsystem we designed consists of two H-bridge motor motor drivers to control the dc shuffling motors and the dealing dc motor along with a single servo motor for rotating the system mounted on top of the base. The dc shuffling motors will run on 6V (provided by the power subsystem and its regulated output) at a constant speed when handling the shuffling. The dealing dc motor will also run on 6V and will perform one high speed rotation at a time to shoot out a single card through the lever arm attached to it. Lastly, the servo motor will be directly connected to the microcontroller and powered by the 6V regulated source from the power subsystem. The L293D H-bridge motor drivers from STelectronics will allow us to control the speed and direction of the 2 shuffling dc motors and the dealing dc motor through PWM signals that it will receive from the microcontroller [2].

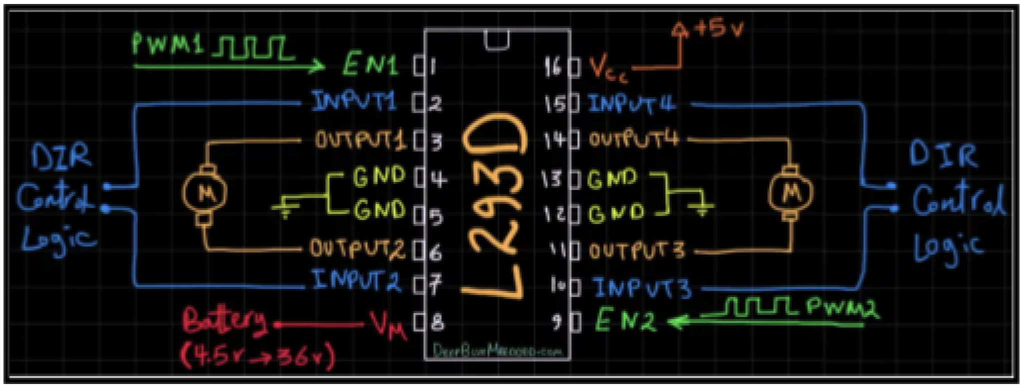

Fig 2.6 *L293D H-Bridge Motor Driver Pinout and control logic*

Each motor driver carries 4 input and output pins respectively and two enable pins allowing for control of two dc motors. The enable pins on the driver will be connected to the microcontroller digital pwm I/O pins and the input pins will be connected to regular digital I/O on the microcontroller. The output pins will be connected to the two terminals of the dc motors. The motor drivers also have a Vcc1 (used to drive the internal logic circuitry) and a Vcc2 (used to power the internal H-bridge of the IC to drive the motors) which will take in 5V and 6V respectively from the power subsystem's regulated outputs. The servo motor simply has a power pin, ground pin, and control pin. The control pin will be connected to a PWM digital I/O pin on the microcontroller such that we can control how much the motor rotates.

3. Verification

In regards to testing and verification of the motor subsystem, I have split up the tests into three main subsections: Microcontroller & Motor Drivers, Shuffling, Dealing, and Rotating. The subsections will involve hardware and/or software testing as they work hand in hand with the STM32 microcontroller chip in order to function properly.

3.1 Microcontroller & Motor Driver Testing:

Test	Verification
Microcontroller: PWM pins must be able to output a stable 0-100% duty cycle PWM wave.	1. Set up our STM32 microcontroller on a breadboard and connect it to a power supply. 2. Connect an oscilloscope probe to the PWM output pins we will be using on the board. Connect the ground clip of the oscilloscope probe to the ground of the microcontroller.

	3. Write a simple program to configure the PWM output pins as an output and set the output to a square wave with duty cycle's 0%, 25%, 50%, 75%, and 100%. 4. Monitor the oscilloscope outputs and ensure they correspond to the programmed duty cycle.
Motor Driver: Each input pin (1, 2EN, 3,4EN, 1A, 2A, 3A, 4A) must switch on and off when connected and given a logic high.	1. Connect the motor driver to a power supply that matches the specified voltage range for the device. 2. Connect the inputs to a logic high voltage source (3.3V). 3. Use a multimeter to verify that the voltage level on the input pins matches the voltage source. 4. Use an oscilloscope to verify that the input pins switch on and off as expected when the logic high voltage is applied. 5. Connect a DC motor to the output pins of the motor driver. 6. Repeat steps 2-4 with the motor driver connected to the motor, to verify that the motor driver outputs the switch on and off as expected when given a logic high.
Motor Driver: The Y1, Y2, Y3, and Y4 pins must be able to run a DC motor with the corresponding enable pin (1,2EN or 3,4EN)..	1. Connect the motor driver to a power supply that matches the specified voltage range for the device. 2. Connect a DC motor to the output pins Y1, Y2, Y3, and Y4 of the motor driver. 3. Connect the corresponding enable pins 1,2EN or 3,4EN to a logic high voltage source (usually 5V or 3.3V). 4. Use a multimeter to verify that the voltage level on the enable pins matches the voltage source. 5. Apply power to the motor driver and verify that the motor runs when the corresponding enable pin is set to high. 6. Use an oscilloscope to verify that the motor output voltage is stable and matches the voltage supplied by the power supply. 7. Measure the current drawn by the motor using a multimeter. Ensure that the current does not exceed the rated current of the motor driver or the motor. 8. Reverse the polarity of the power supply to the motor driver and repeat steps 5-7 to verify that the motor runs in the opposite direction.
Microcontroller: Digital I/O pins for motor driver direction inputs must be able to output a stable 1 or 0 for proper directional control.	1. Connect the microcontroller to a power supply. 2. Connect the digital I/O pins of the microcontroller to the input pins (1A, 2A, 3A, 4A) of the motor driver that control the direction of the motor. 3. Connect a DC motor to the output pins (Y1, Y2, Y3, Y4) of the motor driver. 4. Write a simple test program for the microcontroller that sets the digital I/O pins to output a stable 1 or 0. 5. Upload the test program to the microcontroller and run it. 6. Use a multimeter to verify that the digital I/O pins output a stable 1 or 0, as expected.

	7. Verify that the motor runs in the expected direction when the digital I/O pins are set to the appropriate values. 8. Reverse the polarity of the power supply to the motor driver and repeat steps 4-7 to verify that the motor runs in the opposite direction.

3.2 Shuffling DC Motor Testing:

Test	Verification
Shuffling DC motors must not be defective	1. Connect a 6V power supply to the motor. 2. Connect a switch to the power supply and the motor. 3. Close the switch and observe the motor. 4. If the motor starts spinning, it is not defective. 5. If the motor does not start spinning, double-check the connections and ensure that the power supply is providing a stable 6V. Try reversing the polarity of the power supply and repeat step 3 to see if the motor spins in the opposite direction. 6. If the motor still does not spin or vibrates while spinning, the motor may be defective or damaged. Consider replacing it if necessary.
Shuffling DC motor must be able to output at least 40 oz-in of torque in order to shuffle a standard playing card.	1. Mount the motor onto some support structure (tape to a table for example) and attach a load to the motor shaft. 2. Attach a torque meter. 3. Connect the motor to a variable DC power supply, and set the voltage to 6V. 4. Gradually increase the voltage until the motor reaches its maximum rated current, and record the corresponding torque reading from the torque meter. 5. Repeat step 4 for different voltages, ensuring that the torque readings are consistent. 6. If the motor is able to output at least 40 oz-in of torque, it is considered suitable for shuffling a standard playing card.
Shuffling DC motors must be able to shuffle a full deck of cards (place half the deck on each side of our system) in riffle fashion	1. Prepare the mechanical enclosure with the motors installed within them properly. 2. Place half the deck of cards on each side of the system, ensuring that they are evenly distributed. 3. Connect the motors to a suitable power supply and controller that will allow you to control the speed and direction of the motors.

(one card from the left motor, next from the right) and mostly evenly.	4. Activate the motors and observe the shuffling process, ensuring that the cards are shuffled in riffle fashion (one card from the left motor, next from the right) and that the cards are mostly evenly distributed. 5. Repeat the shuffling process multiple times, and ensure that the results are consistent and repeatable. 6. If the motors are unable to shuffle the cards in the desired fashion or if the cards are not mostly evenly distributed, try adjusting the motor speed, direction, or position within the system.

3.3 Dealing DC Motor Testing:

Test	Verification
Dealing DC Motor must not be defective	1. Connect a 6V power supply to the motor. 2. Connect a switch to the power supply and the motor. 3. Close the switch and observe the motor. 4. If the motor starts spinning, it is not defective. 5. If the motor does not start spinning, double-check the connections and ensure that the power supply is providing a stable 6V. Try reversing the polarity of the power supply and repeating step 3 to see if the motor spins in the opposite direction. 6. If the motor still does not spin or vibrates while spinning, the motor may be defective or damaged. Consider replacing it if necessary.
Dealing DC motor needs at least 50 oz-in of torque in order to successfully launch a standard playing card (weighs roughly 100-150g) to a distance of 1 meter.	1. Mount the motor with a load onto some surface (table with table again for example). 2. Attach a torque meter. 3. Connect the motor to a variable DC power supply, and set the voltage to 6V. 4. Gradually increase the voltage until the motor reaches its maximum rated current, and record the corresponding torque reading from a torque meter attached to the motor shaft. 5. If the motor is able to output at least 55 oz-in of torque, it is considered suitable for dealing out a single standard playing card.
Dealing DC motor must be able to shoot out 1 card at a time using the arm level attached to it and the pwm signals from the h-bridge motor driver	1. Prepare the mechanical enclosure with the motor properly installed in it. 2. Connect the DC motor to the motor driver and to a power supply to control the motor. 3. Place a single card into the card holder mechanism that is connected to the arm level. 4. Activate the DC motor and observe the card shooting out of the card holder mechanism, ensuring that only one card is shot out at a time.

	5. Repeat the process multiple times and ensure that the results are consistent. 6. If the DC motor is unable to shoot out only one card at a time, try adjusting the PWM signals or arm level position.

3.4 Rotating Servo Motor Testing:

Test	Verification
Rotation Servo Motor must not be defective	1. Connect a 6V power supply to the servo motor. 2. Connect the signal wire of the servo motor to a PWM output pin of a microcontroller. 3. Write a simple program for the microcontroller to send a PWM signal to the servo motor, commanding it to rotate to 90 degrees. 4. Observe the servo motor and ensure that it rotates to the commanded angle smoothly and accurately. 5. If the servo motor does not rotate or rotates erratically, double-check the connections and ensure that the power supply is providing a stable 6V. 6. If the servo motor still does not function properly, it may be defective or damaged. Consider replacing it if necessary.
Rotation servo motor must output enough torque (>80 oz-in) to rotate the device	1. Mount the servo motor onto some surface (table with table). 2. Connect a 6V power supply to the servo motor. 3. Connect the signal wire of the servo motor to a PWM output pin of a microcontroller. 4. Write a simple program for the microcontroller to send a PWM signal to the servo motor, commanding it to rotate 180 degrees. 5. Slowly increase the load torque on the motor shaft using a torque wrench gradually increase the torque in increments of 10 oz-in while monitoring the motor's response. 6. At each torque level, observe the motor's ability to reach the 180 degree rotation angle target. Ensure that the motor is not struggling or vibrating excessively, which can indicate insufficient torque output. 7. If the motor is able to maintain the target angle at a torque level of at least 80 oz-in, then it is strong enough to rotate our whole device.
Must have the capability to rotate at least 180 in increments of 11.25, 12.9, 15, 18, 22.5, 30, 45, and 90	1. Connect the servo motor to a 6V power supply and to the microcontroller. 2. Write and upload a simple program to the microcontroller to rotate the motor at a specified angle. 3. Set the servo motor to its initial position (usually 0 degrees). 4. Using the protractor, verify that the servo motor is correctly positioned

degrees (1-8 players) in a controlled manner.	at 0 degrees. 5. Test the servo motor's ability to rotate in increments of 11.25, 12.9, 15, 18, 22.5, 30, 45, and 90 degrees. For each increment, do the following: a. Set the servo motor to the desired angle in the microcontroller program. b. Use the protractor to verify that the servo motor has moved to the correct angle. 6. Verify that the servo motor moves smoothly and without any jerky or sudden movements. 7. Record the results and ensure that the servo motor can rotate properly by all of the increments mentioned above.

4. Conclusion

4.1 Self-Assessment

Currently, we are on track with our tasks and assignments as a team and individually. As mentioned in the introduction, the initial designing pertaining to the project was done primarily as a group, so the workload was quite evenly distributed. With splitting up testing, verification, and our tasks for the remainder of the semester, we have done a good job dividing the workload evenly.

4.2 Timeline/Plan:

For the rest of the semester we have created a timeline and schedule as a team and individually to complete our project. I will first continue with microcontroller development and breadboard testing until the next round of PCB orders and make any necessary changes prior to it. Around the time of the team contract fulfillment, I will have a working prototype of the shuffling, dealing, and rotation sections of the motor subsystem and will debug both on the software and hardware end. Afterwards we will put all the components of the project together as a team and add any extra functionality with the time we have left. Finally, we will prepare for our final demo and begin working on the final presentation and paper.

03/20 - 03/26	Microcontroller Code Development and Motor Subsystem Breadboard Testing (shuffling dc motors, H-bridge motor drivers, communication with microcontroller pins)

03/27 - 04/02	PCB Edits, Third Round PCB Orders, Individual Progress Reports, further Motor Subsystem Breadboard Testing (dealing dc motor with lever and H-bridge motor driver, servo motor for rotation, communication with microcontroller pins)
04/03 - 04/09	MVP, basic shuffling and dealing functionality implemented, Final PCB Adjustments and Orders
04/10 - 04/16	Team Contract Fulfillment, Debug Motor Subsystem and microcontroller code
04/17 - 04/23	Mock Demo Link all components of the project together, Add additional functionality
04/24 - 04/31	Debug any last minute things as a group, Final Demo, Final Review of Presentation and Paper
5/1 ++	Final Presentation, Final Paper

4.3 Ethical Considerations:

With developing and testing this motor subsystem, there are many ethical considerations:

1. *[1,2] IEEE Code of Ethics and ACM Code of Ethics (Integrity):* Upholding highest standards of integrity, responsible behavior, and ethical conduct in professional activities.
    a. The motor system will be designed and implemented in a way that ensures honesty and transparency in the shuffling and dealing process (card shuffling process is visible to the players). The code used to run the motors will produce truly random results with the final card shuffle; the users will be able to detect any attempts to manipulate the system.
    b. Since the motor subsystem and how it is mounted onto our project is quite mechanical, I will frequently meet with the Machine Shop Technicians to make sure none of the motors and any design changes can be hazardous to a user.

2. *[1,2] IEEE Code of Ethics and ACM Code of Ethics (Competence)*: The designer and developer must have the necessary skills and knowledge to implement the system.
    a. I have experience working with h-bridge motor drivers both from the hardware and software side; I have worked on various projects involving motor subsystems including a gimbal to mount a lidar with pitch and yaw control, and rotation of the platform.
    b. I will consult with teaching assistants, professors, online sources, and any other sources available to me when making design decisions, connecting components, and programming the subsystem to ensure safe functionality and fairness.

3. *IEEE Code of Ethics and ACM Code of Ethics (Professionalism):* To strive to ensure this code is upheld by colleagues and co-workers.
    a. As a team we have agreed to meet up on a weekly basis and have been doing so to ensure we are all on track with our tasks and schedules. These meetings will also cover any

violations of the IEEE and ACM codes that surface throughout the semester. If needed we will bring the matter up to professors and/or teaching assistants to handle it properly.

5. Citations

[1] “IEEE Code of Ethics.” *IEEE*, June 2020, https://www.ieee.org/about/corporate/governance/p7-8.html

[2]"STM32 DC Motor Speed Control using PWM & L293D Examples." Deep Blue Embedded. Accessed March 29, 2023. https://deepbluembedded.com/stm32-dc-motor-speed-control-pwm-l293d-examples/.

[3]ACM, the Association for Computing Machinery, has a code of ethics for computing professionals. The website for the code of ethics is: https://www.acm.org/code-of-ethics.