ECE 445 SP23
Design Document
Autonomous Card Dealer

Adam Naboulsi, Rohit Chalamala, Ralph Balita

February 23, 2023

Table of Contents

1. Introduction

a. Problem and Solution

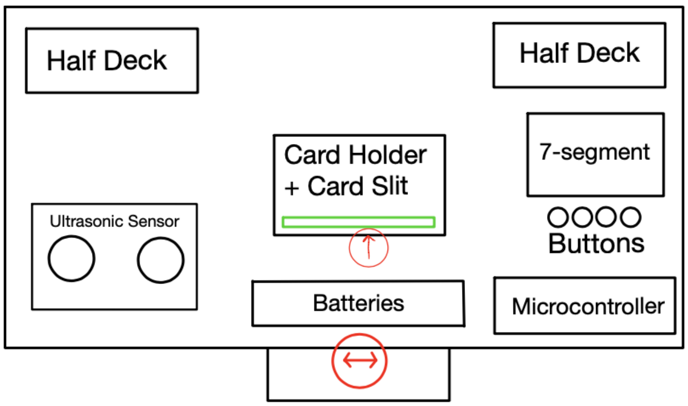
We all love card games, whether we are just playing casually with our friends or going to the casino. To name a few: Poker, Literature, Blackjack, Kings Corner, etc. Our goal is to create a fair card shuffling and distribution system in which the gameplay is smooth, effortless and eliminates the possibility of cheating. To add complexity, our group plans on implementing programmable game modes including and/or not limited to the ones named above. In these cases, we plan on programming the bot to deal cards to the players and set up the card game playing field.

At a high level, we want to make the card game playing process more effortless and fair by replacing the dealer with a device. There are a few different subsystems of this project including the card shuffler, card dealer/distributor, and the user interface. The card shuffler allows the games to be more fair because it gets rid of human error that is often present with shuffling. The card distributor would basically replace the dealer by being able to rotate and shoot out cards to certain locations on the board. The user interface for this would be some kind of buttons that allows the user to control turning the device on and off, the number of players, the game mode, and starting/pausing the game, etc. This will solve the problems of current human dealers by making the whole process a lot more fair.

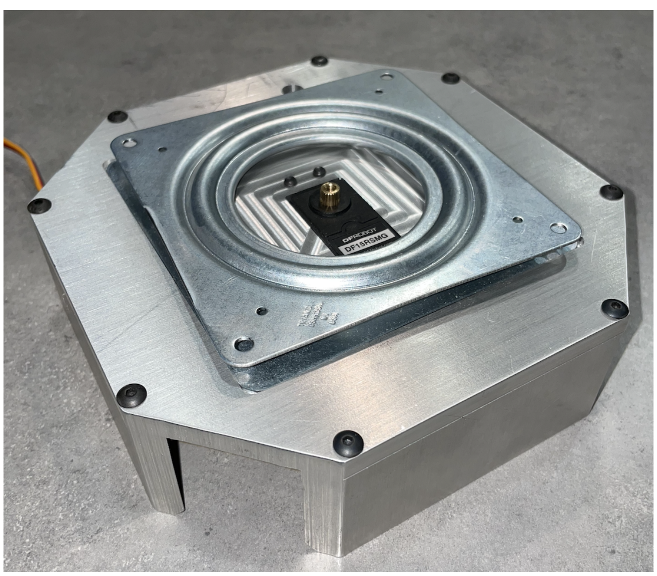
Summary: We are building an autonomous card shuffler and dealer for casual card playing. This can be used by our customers to play various games such as Poker, Blackjack, Kings Corner, etc. Many players have a problem with annoyance in dealing and potentially having an unfair dealer. We plan to automate the dealing aspect of the game by having the cards launched to the players and setting up the playing field for the game. As a result, our customers can enjoy playing the game in a fair and effortless manner.

b. Visual Aid

1) This is the card shuffler that we have bought online. The only reason we actually need this is to reduce the mechanical complexity for the machine shop, however, we will be reusing their motors and gears so as to not waste any part. But we will be doing all of the electrical work and scrapping any electrical work they might already have. In the card holder area is where the launch system will go.



2) This is the approximate front view of how the system would look after we add in the dealing components. We will be adding the ultrasonic sensor and 7-segment display to help us achieve our design. The card holder from the initial machine will be modified so that we have a slit that only 1 card can use, and we can have a motor that will launch out the cards. This will all be connected to the rotating base.



3) This is the rotating base that we will be using, we will mount the rest of our project on top of this base. This was already made by the machine shop for a previous project, and we determined a way to make it work for our project. Since the machine shop won't have to rebuild a base, it will make it easier for them. There is also a servo that is already attached to this base, so we will use this so we aren't wasting parts.

All the components of this visual aid work together to shuffle and distribute cards autonomously.

c. High-level requirements list

a. Shuffle a set of cards evenly

i. Given any two sub-decks, the shuffling system shall perform one riffle shuffle. The result of the riffle shuffle shall be a deck of cards that are aligned and ready to be distributed. No card left behind.

ii. One test that our group plans to perform and test the riffle shuffle mechanism involves two sub-decks. i.e. *13 in-ascending-order cards Spades suited and 13 in-ascending-order cards Hearts suited*. The test will be noted as a **SUCCESS** if the result of the shuffling mechanism is shuffled any order of *As-Ah-2s-2h-3s-3h-4s-4h-5s-5h-6s-6h-7s-7h -8s-8h-9s-9h-Ts-Th-Js-J-Qs-Qh-Ks-Kh*. The test will be noted as a **FAILURE** if the resulting deck does not include all of the cards (some cards were left behind). We will analyze the effective-ness of the mechanism by comparing the order of the sub-decks before and the resulting pile of cards after the card shuffle is done.

b. Distribute the cards to the players and Set up playing field

i. Given an aligned and ready to be distributed deck of cards, the distribution system shall perform a dealing mechanism. The dealing mechanism shall pick up the cards one at a time and deal the card towards its intended trajectory. This 'dealing' action is performed until the deck has been depleted.

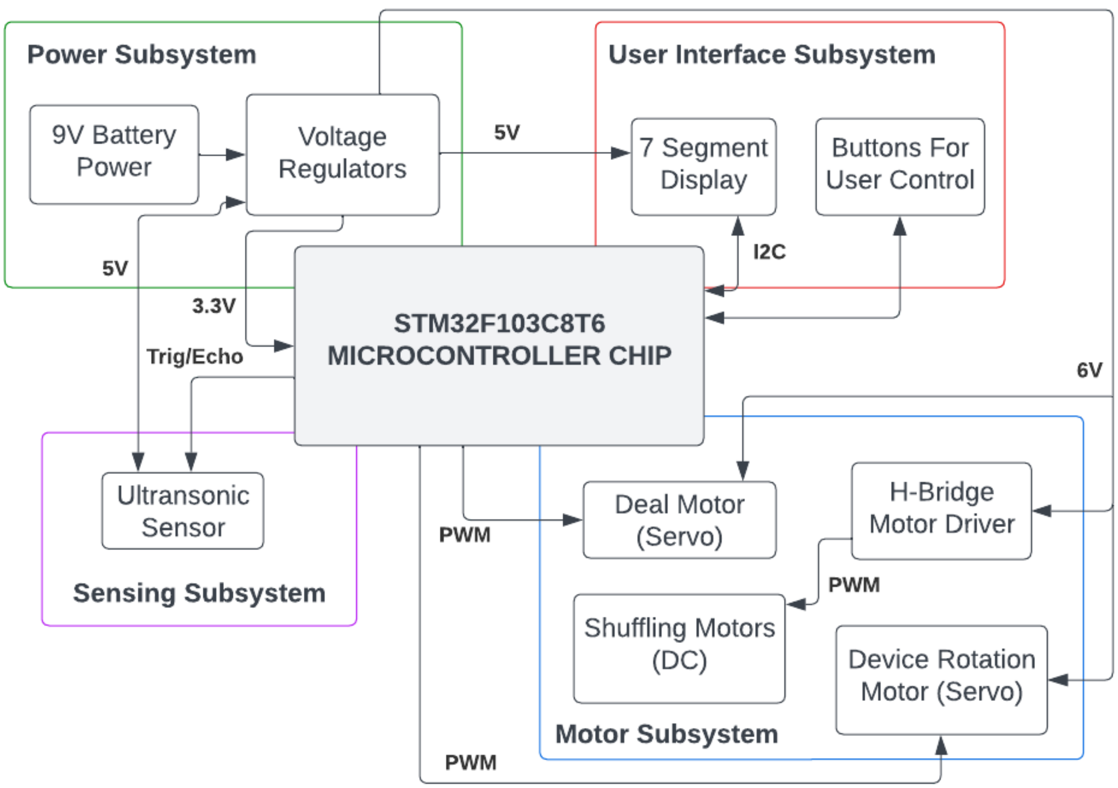
ii. The first test that our group plans to perform in order to test the card distribution system involves one deck of unshuffled cards and preset modes. i.e. number of players = 2, and game mode = evenly distributed amongst all players. The test will be noted as a **SUCCESS** if the number of cards dealt to each player is the same (26 each). The test will be noted as a **FAILURE** if not all cards are dealt OR if the number of cards dealt to each player is **NOT** the same. The distribution system shall rotate to a preset 'next' angular position, stop rotating, and deal a card. The system shall continue to perform this act for all preset 'next' angular positions (0,90,180) until the deck is depleted. Our group plans on generating more tests that involve various numbers of players (4,8) which are restricted by the game mode. Overall, this system can be tested visually by checking that the cards that are on the board have been distributed correctly and in the right order.

c. A functioning user interface with a display and buttons for parameter adjustment

i. Our goal is to abstract the device to work on multiple game modes and various numbers of players. The user interface shall involve 4 buttons. On/Off, Start/Pause, toggle number of players, and toggle game mode. It will also have a 7-segment display that shows the user the current setting for number of players, and game mode (numbered). This high level requirement is software heavy, given that the various game modes restrict the number of players. This, in turn, will impact how the deck is dealt. The test will be noted as a **SUCCESS** if the device deals the correct piles of cards to the correct number of players for a given game mode and num_players. The test will be noted as a **FAILURE** if the dealing sequence fails to match the request of the user. Test each of the buttons to make sure they are performing as expected and updating the displayed game settings correctly.

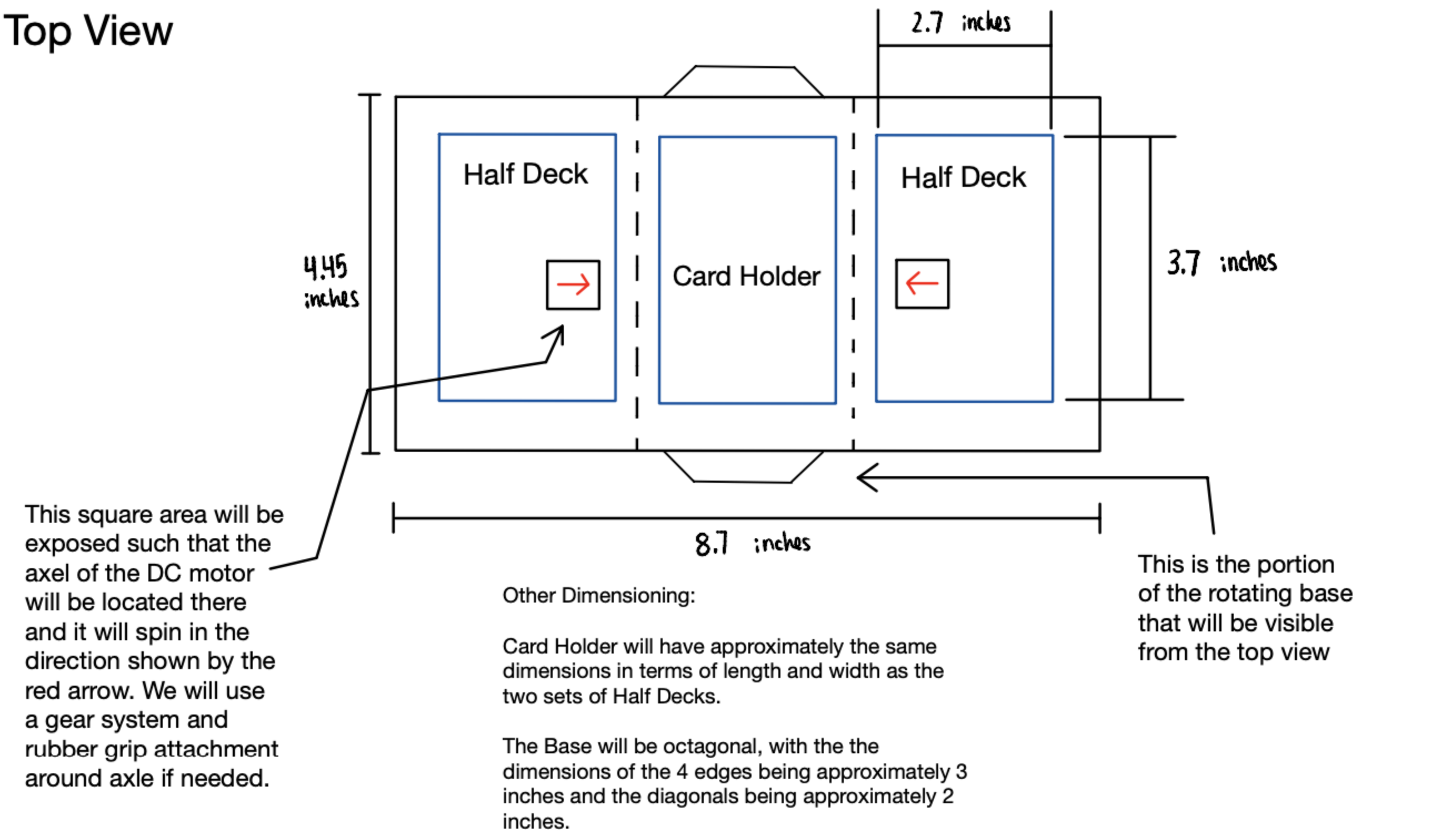
2. Design

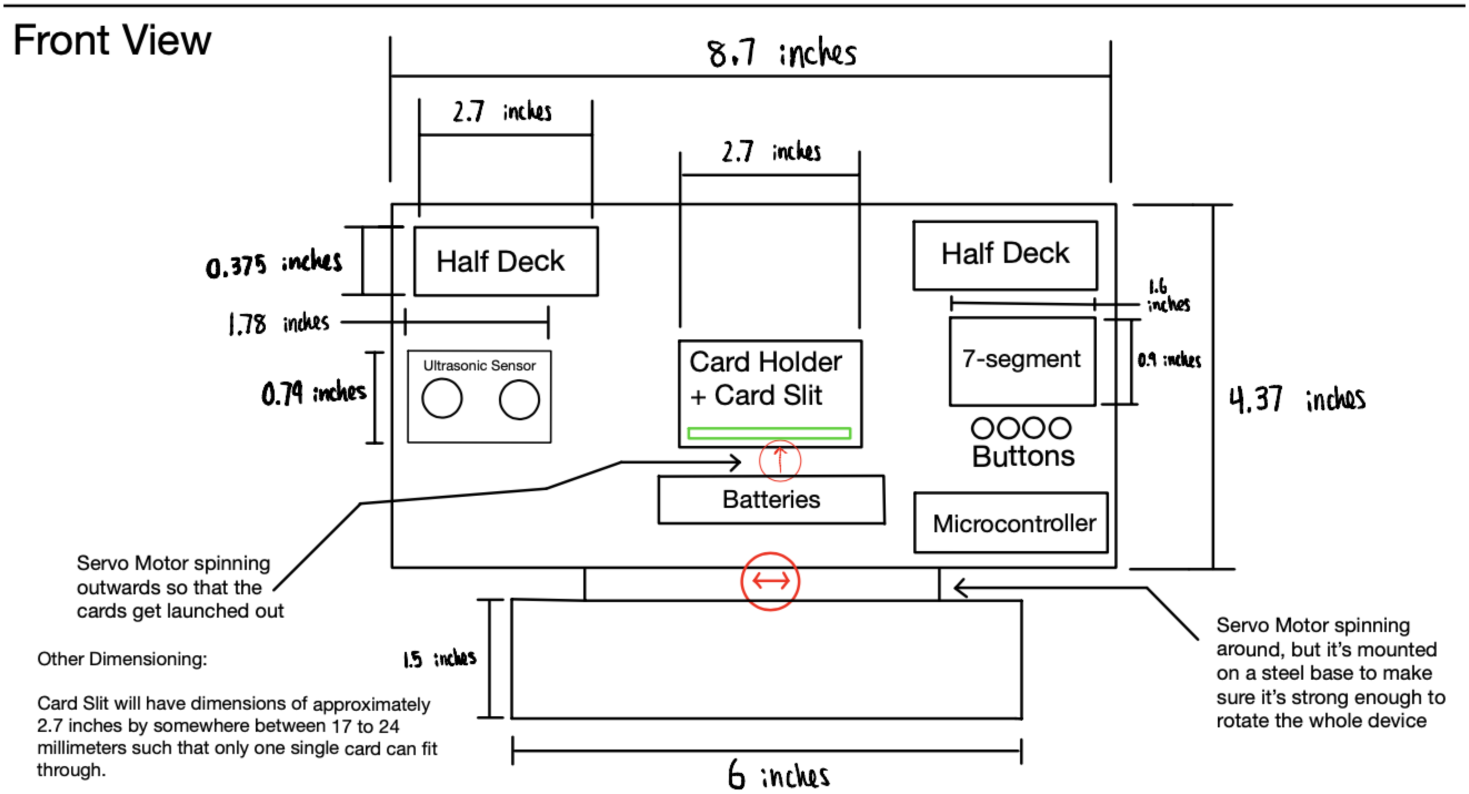
a. Block Diagram



There are 4 main subsystems that we have in our design and block diagram. They are the sensing subsystem, power subsystem, user interface subsystem, and the motor subsystem. These subsystems will be interfacing with each other and also with our STM32F103C8T6 Microcontroller Chip. The power subsystem will power our microcontroller and motor subsystem, and also indirectly power our sensing and user interface subsystems through the microcontroller. The sensing subsystem will consist of an ultrasonic sensor that will read when a player is in front of our device so that we can launch a card to them. This is where the motor subsystem comes in, it will be responsible for rotating the base so the cards are spread across the table, it will also be launching the card, and also it will be responsible for shuffling the cards. Lastly, there is the user interface subsystem which will be buttons and a seven segment display, the buttons will be used to determine game mode and number of players and the seven segment display will accurately represent this. All of these components together will allow us to have a system that can shuffle cards and then deal them as well.

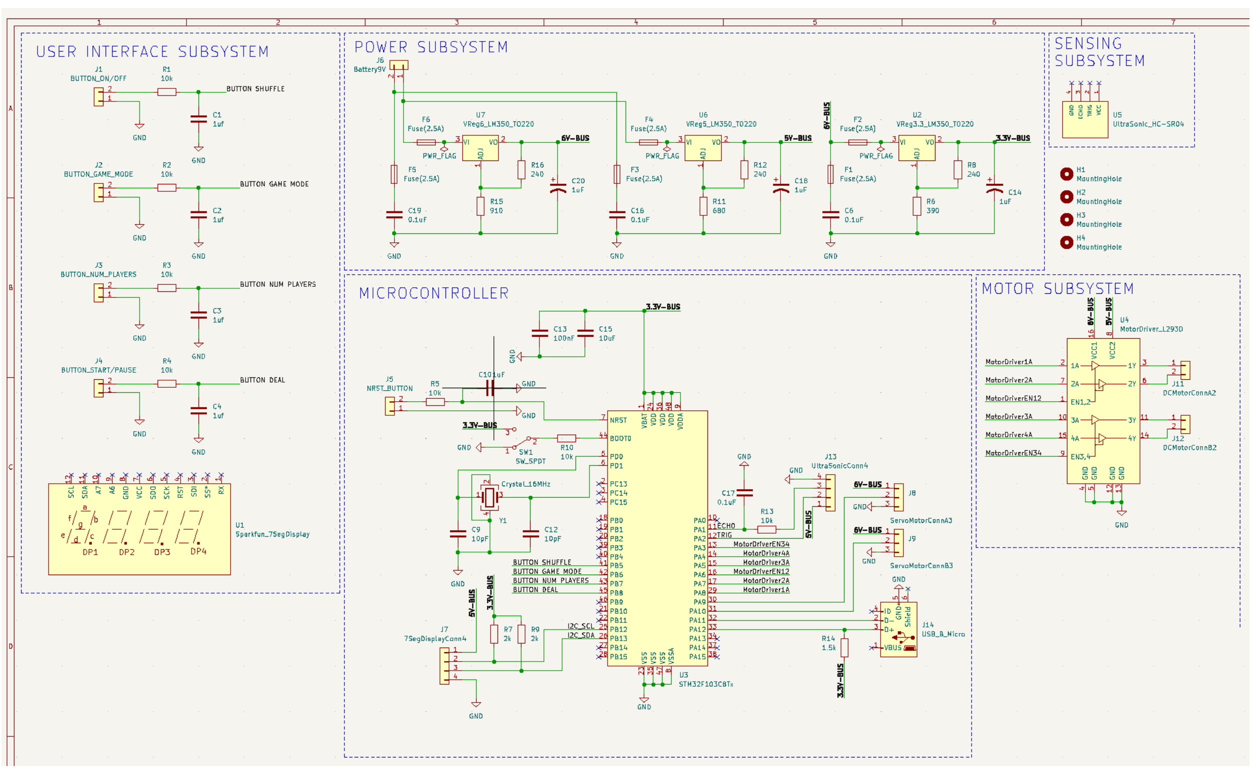
b. Physical Design





The above design shows both a top and a front view. The side view is unnecessary as it doesn't provide any additional useful information. In this design we have a rotating base which will be able to rotate 180 degrees in order to distribute our cards. On top of that we have our card shuffler and dealer mechanism. The card shuffler uses two dc motors to shuffle the cards one by one into the card holder. At this point, the ultrasonic sensor will determine where the players are while the base is rotating and launch cards at each point. The cards will be launched through the use of a servo motor which will launch them through a card slit that is just large enough to fit a singular card. We will also have our 7-segment display which will display the number of players and the current game mode. Along with this we will have buttons that allow us to select our desired settings. Lastly, our microcontroller will be in the enclosure as well. **Note:** For the subsystems described below, we will be using an STM32F103C8T6 Microcontroller Chip to interface between all of them in our device.

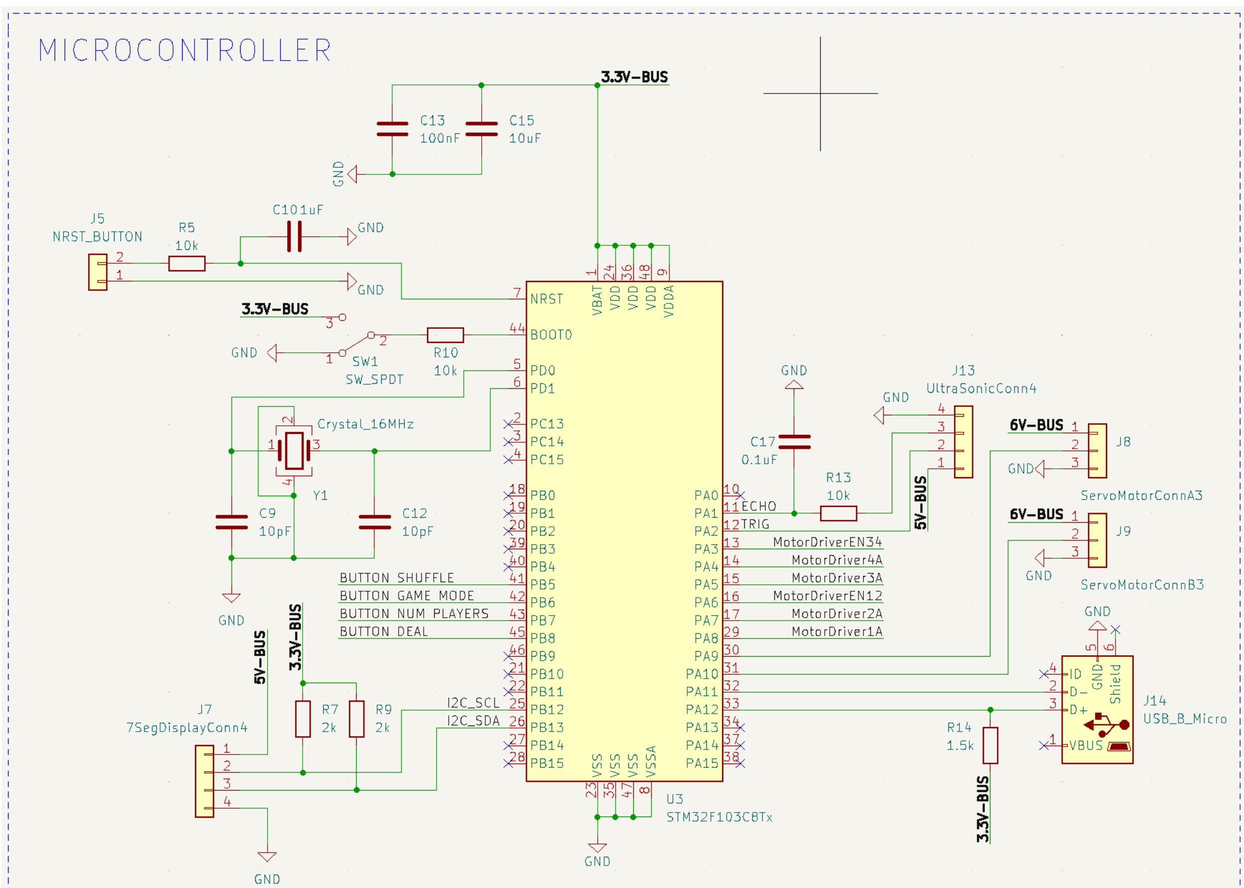
Schematic:



c. Microcontroller

Microcontroller:

Although this is not considered as a subsystem for this project, it is important to understand how the microcontroller interfaces with and between the devices of the various subsystems. As you can see in the schematic below, the top left represents necessary parts to power and control the microcontroller (buttons, voltage input, clock). The bottom left of the diagram presents the pins of the microcontroller that will be used for user interfaces, including buttons and the 7 segment display using I2C. On the top right, you can see how the Ultrasonic Sensor (our sensing subsystem) relies on an echo and trigger pin from the Microcontroller. The middle right represents the interface between the motor subsystem and the microcontroller, including the servo motor connection and the Motor Driver pins. Lastly, the bottom right represents the USB-B Micro for programming, this will be used to update the program running on the microcontroller as we test. Include bypass capacitors



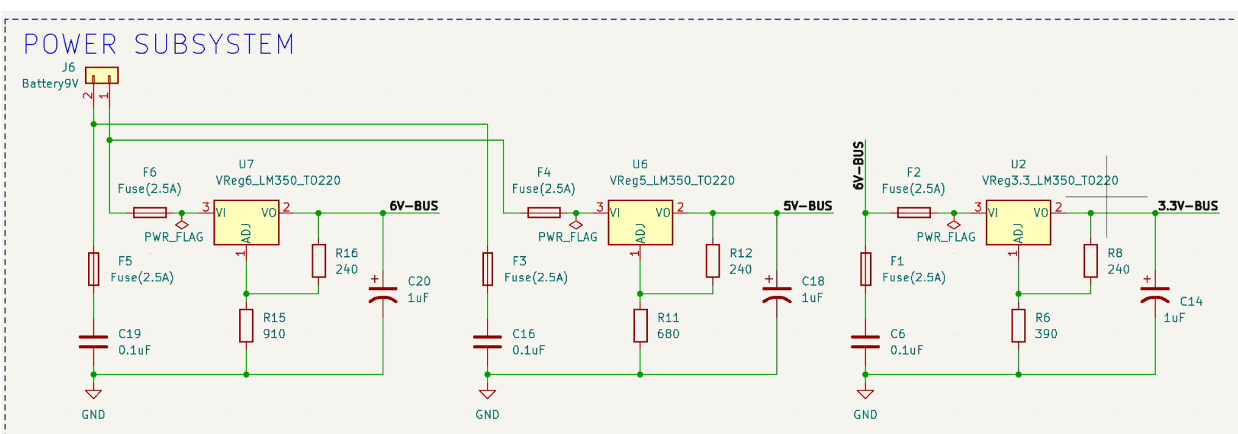
d. Power Subsystem

Power Subsystem:

The function of this system is to power all of the other subsystems in our device, regulate the voltage supplied to the motors – ensuring a stable rpm –, reduce electrical noise in the output, and provide some circuit protection. It will be powered by a 9V battery through a clip connector and provide stable 6V, 5V, and 3.3V outputs in order to make sure the microcontroller, motors, chips, and sensors are powered sufficiently. The 9V battery will feed into two LM350 voltage regulators which will output 5V and 6V respectively. The 6V output will then feed into another LM350 voltage regulator to output 3.3V as well. These LM350 voltage regulators can withstand a 3A load current (the highest operating current consumed by our devices is 2.5A). Lastly, we will have fuses connected to the voltage regulators with a current limit of 2.5A for short-circuit protection.

Requirements	**Verification**
Must provide a stable, regulated 6 ± 0.1V, 5 ± 0.1V, and 3.3V ± 0.1V in order to power the various parts of our project.	1. We will use a voltmeter at the output of each voltage regulator to make sure that they are outputting their intended outputs (a constant 6 ± 0.1 V, 5 ± 0.1 V, and 3.3 ± 0.1 V correspondingly). We will also test at the inputs of all of our motors, sensors, and chips to make sure the voltage is stable. 2. This will be shown in our lab notebook by writing down voltage at each of the locations that was discussed above.
Must support up to 2.5A of current drawn to prevent overheating caused by our servo motor.	1. Our maximum amperage is from our motors. We also got a voltage regulator that will be able to deal with this level of current. 2. In order to make sure our power subsystem is capable of dealing with the 2.5 A of current from the motors, we will first use a multimeter to measure the current output from each of the motors. Then we will add up all of these currents to make sure that it is less than the 2.5 A that our voltage regulator is capable of using. 3. If it is larger than this, we can add in some resistors, since V = IR, if we increase resistance, current will go down. We will calculate these resistances and then measure the current again with the multimeter. 4. This will be shown in our lab notebook by discussing if we had to use any resistors, and what the current usage of each of our components was, and then writing down the

	total current consumed by our system.
Must provide short-circuit protection to prevent any components from being damaged in the case where too much current is drawn by our motors.	1. The first step to ensuring that the project doesn't short circuit is to take a thorough look through the design and look for any flaws prior to starting building. In specific it is important to look at our schematic along with the components we ordered to make sure they can provide what we need. 2. We will make sure all of our wires aren't tangled or exposed and follow any other electrical safety guidelines. 3. We will also make sure to add fuses which should help in preventing any short circuits from occurring. 4. In order to actually test this, we will use a multimeter and measure the currents and voltages at every possible point in our circuit to make sure everything is as expected. 5. We will show this in our lab notebook, by noting down all of our measurements, and stating if there are any measurements that we feel are off, so that we can discuss those and understand why they might be like that.



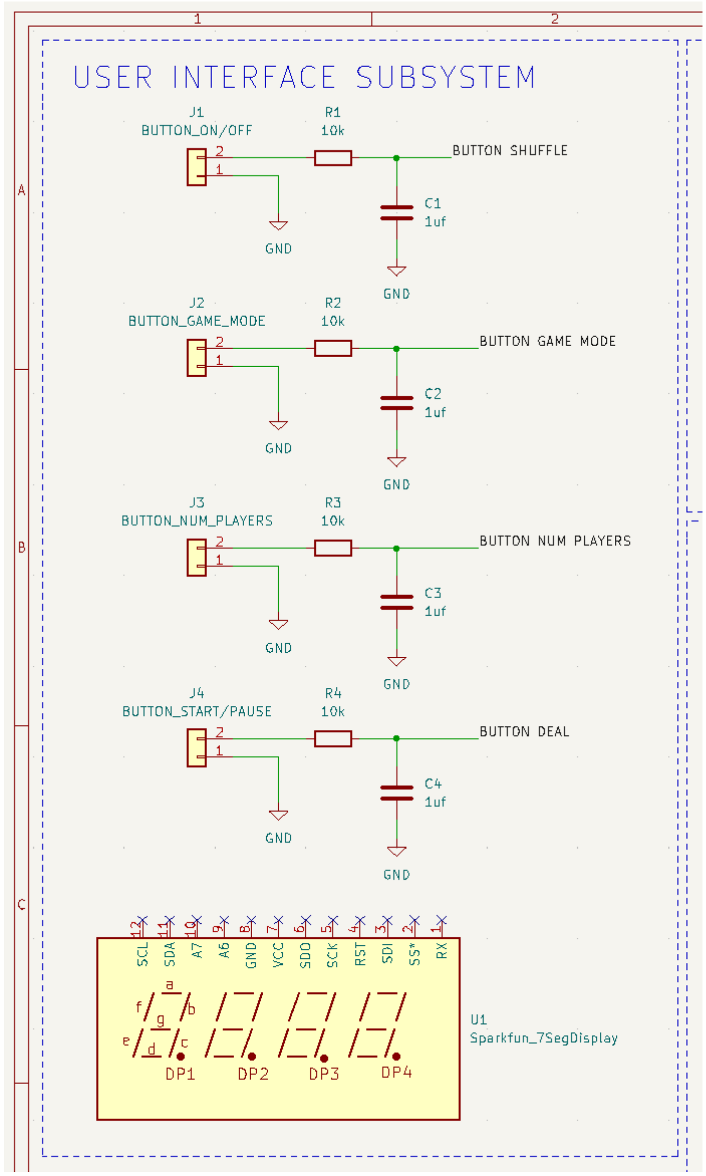
e. User Interface Subsystem

User Interface Subsystem:

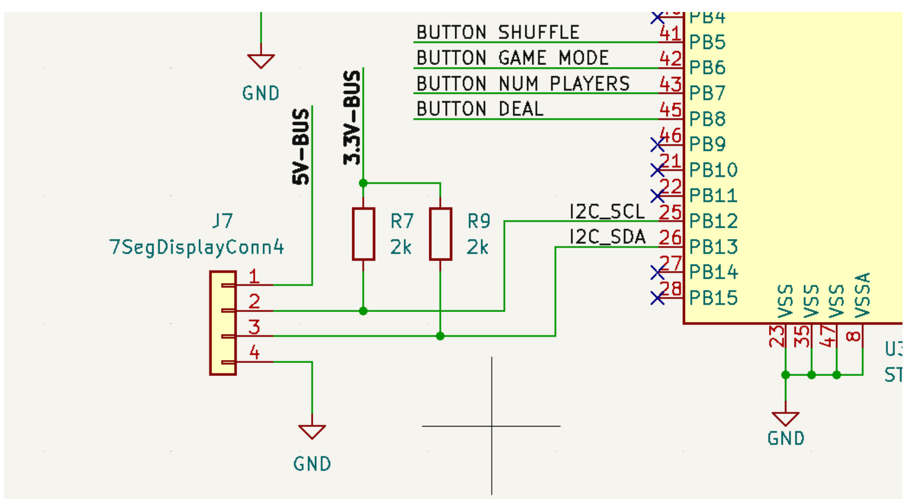
The function of this subsystem is to allow the user to be able to have control over our device. The user will be able to select the number of players and the game mode, and either shuffle or deal the cards through clicking buttons. The current game mode and number of players will be shown on the 7-segment display. The buttons will be labeled with their corresponding functions on the device and will indicate what values to change or begin displaying on the 7-segment display which will receive instructions from the microcontroller through SPI. The 7-Segment display we are planning to use allows the user to pick between SPI, I2C, and RST communication protocols. Since there is only one display to communicate with and we do not need to be very conservative on resources, we decided to use I2C as the communication protocol between the microcontroller and the display.

Requirements	Verification
Must be able to indicate the number of players (1-8) through the pressing of a button that increments. Must also be able to change the game mode (Poker, Literature, Blackjack, and King's Corner) through also pressing an incrementing button.	1. This will be mostly done in software, we will need to test and debug our code to make sure that it is outputting the correct values. 2. The first way to verify this is by pressing the buttons and checking if we are getting the specified output in software. 3. Once we have hooked up the 7 segment display, we can also visually verify that every time the button is pressed, the values change and output the desired number of players/the game mode. 4. The last verification mode is to actually run the whole device and make sure that it is outputting the cards to the correct number of players, and in the right way for the selected game mode. 5. We will show this in our lab notebook by listing out what occurs each time we press the button and determine if it changes the settings in the correct manner. We will also state if there were any errors and how we can fix those.
Must communicate display data (number of players and game mode) to microcontroller using I2C protocol.	1. We are using the I2C to communicate, we will be able to mostly do software to make sure this is working and to validate it. 2. In order to verify that the signals are correct, we can use print statements to print out what will be communicated from the microcontroller to the 7-segment display through I2C.

	3. Next, we can verify visually by checking that what is communicated to the 7-segment display through I2C is actually getting reflected on the display. We want to make sure it is getting updated every time that the button is pressed, so if it's not we will have to debug. 4. We will write this in our lab notebook by discussing what we are outputting from our microcontroller through I2C and what actually showed up on the display, and then writing down if it's accurate or not.
Must be able to display two hex digits with enough brightness on 5V operating voltage.	1. The 7-segment display should be able to work with only 5V. The microcontroller has a 5V pin that we will be wiring to our 7-segment display to power it. 2. We will validate that this works by checking if the 7-segment display turns on when we power it on using just the 5 volts. 3. We will also validate that the power is consistent and correct to the 7-segment display through the use of a voltmeter. 4. We will write in our lab notebook by writing if our 7-segment display is working correctly with 5 volts of input. We will also discuss how consistent the readings we got from the voltmeter are.
Must have working buttons to change current game mode (Poker, Literature, Blackjack, and King's Corner), change number of players (1-8), initiate shuffle (on/off), and deal process (on/off).	1. We will have 4 buttons that can be used and displayed on our 7-segment display. 2. We will verify that all of the signals that we are getting from these buttons are accurate by printing them out in software to make sure we output the correct values to the 7-segment display. 3. We will verify that the results that show up on the 7-segment display are correct relative to how many times the buttons were pressed. 4. We will also verify visually just by checking that the device operates properly and deals to the correct number of players, uses the right game mode, shuffles when we press the button, and deals when we press the other button. 5. We will write in our notebook regarding anything that went wrong. We will also write about if the buttons correspond to the actions and display output as we expected.



USER INTERFACE SUBSYSTEM
J1
BUTTON_ON/OFF
R1
10k
BUTTON SHUFFLE
C1
1uf
GND
J2
BUTTON_GAME_MODE
R2
10k
BUTTON GAME MODE
C2
1uf
J3
BUTTON_NUM_PLAYERS
R3
10k
BUTTON NUM PLAYERS
C3
1uf
J4
BUTTON_START/PAUSE
R4
10k
BUTTON DEAL
C4
1uf
SCL SDA A7 A6 GND VCC SDO SCK RST SDI SS* RX
DP1 DP2 DP3 DP4
U1
Sparkfun_7SegDisplay



BUTTON SHUFFLE
BUTTON GAME MODE
BUTTON NUM PLAYERS
BUTTON DEAL
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
GND
5V-BUS
3.3V-BUS
J7
7SegDisplayConn4
R7
2k
R9
2k
I2C_SCL
I2C_SDA
VSS VSS VSS VSSA

f. Sensing Subsystem

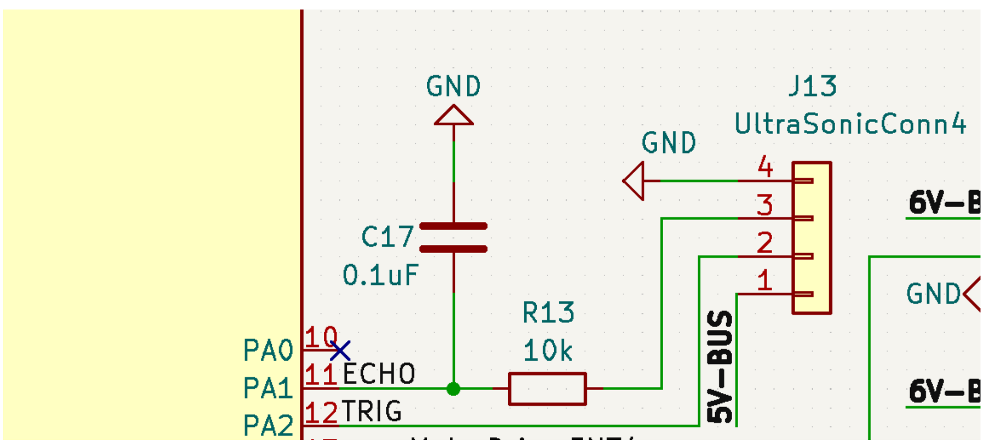
Sensing Subsystem:

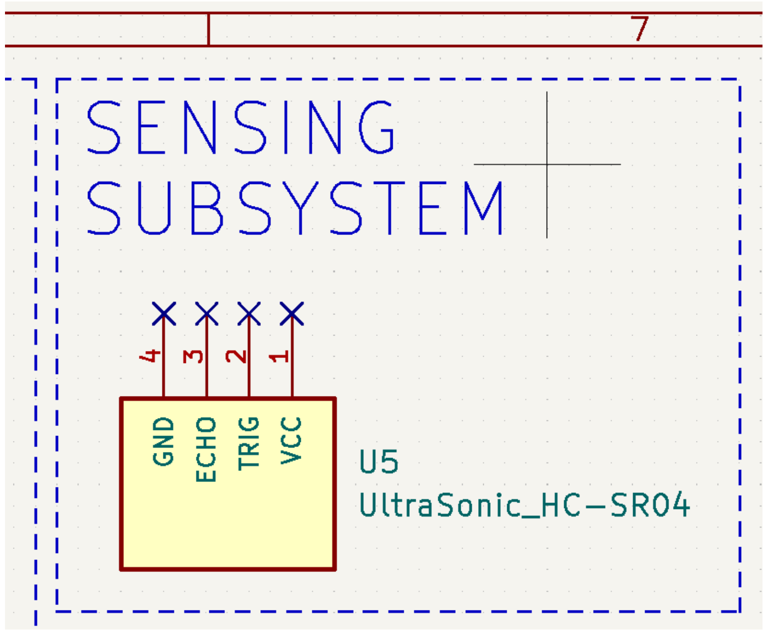
The purpose of this subsystem is to correctly identify how far away each player is from the device in order to ensure that the cards are dealt close to the players on the table. This will be done by collecting readings from an ultrasonic sensor such that when a sound wave hits a player, the device knows roughly where to deal the card. This information will then be used to adjust the speed of the servo motor in charge of launching the cards detailed further below to ensure it lands closer to the player on the table. The ultrasonic sensor we are planning to use uses its own communication protocol involving trigger and echo pins which will be the mechanism our microcontroller uses to collect readings from the ultrasonic sensor.

Requirements	Verification
Must be mounted on device such that the waves hit the people and bounce back	1. We will mount it at a correct angle so that it will hit the players no matter their height. 2. We will verify that this is working by testing with a group of people. We will have people sit around a table, and everytime we move the ultrasonic sensor across a player, we should get some feedback. We will make sure we are getting some readings. 3. We will write in our lab notebook about if when we move across a player, we get a reading or not. If we are not getting a reading, there might be something wrong with our mounting or our code.
Must get accurate readings so that the device will know when to start/stop dealing cards as it rotates around the table from player to player.	1. We will verify firstly that we are getting some type of readings from the ultrasonic sensor when we move across players. 2. We can also print out the readings that we are getting from the players to see if they are accurate and reflect how far away the players are. We want to make sure that we are accurately capturing distances so we know how far to launch the cards. 3. Whenever we get a reading, we want to stop the rotation and launch out a card, we can visually verify that this is occurring. 4. We will write the readings that we get in our notebook. We will see if the readings are accurate in regards to the distance of the player. If they aren't accurate, we might have to adjust the speed of the motors for which the card is being

	launched so that we are launching the cards the correct distance.
Must be able to get readings from at least 2 meters away as that will ensure that all players at the table will be seen and be dealt cards.	1. We want this to be possible because some tables are a lot longer, so we need to account for this and make sure we can tell how far the players are, even if they are at the edge of the table. 2. This can be verified by having a player stand 2 meters away from the ultrasonic sensor, and then running the sensor and checking if we get any readings. If we get some type of reading, this means that it can read things that are 2 meters away. 3. The next thing that we will have to verify is that the readings are actually accurate and tell us that the player is 2 meters away. 4. This can also be verified visually once our device is working by checking that a player who is 2 meters away can get their card distributed to them. 5. In our notebook, we will write more about our findings from the ultrasonic readings. We will write about how well it reads from 2 meters away, and what the max range might be.
We want the measure angle for this sensor to be around 10-15 degrees wide so that the sensor is only focused on one singular player at a time, so we will get more accurate readings.	1. We will test that our ultrasonic sensor has a measure angle this wide. We need it to be only this wide so that we get more accurate readings since each person will probably only be a few degrees wide depending how far away from the table they are. 2. We can verify this by having the ultrasonic sensor point forwards, and testing out placing objects in various positions and checking if they can be read or not. For example, you have one object in the middle, and then every 5 degrees to the left and right of that middle object, you have another object. This would allow you to test where the ultrasonic sensor is no longer able to read the object. 3. In our notebook, we will write the results of this simulation, and check what the maximum measure angle of our ultrasonic sensor is.

Must be able to collect and transmit readings to the microcontroller with a range of at least 2m on 5V operating voltage.	1. The ultrasonic sensor should be able to operate on just 5 volts. The microcontroller has a 5V pin that we can wire to the ultrasonic sensor in order to power it. 2. We will validate that this works by checking if we can get readings from our ultrasonic sensor when we power it using just those 5 volts from the microcontroller. 3. We will also validate that the power is correct and reliable by using a voltmeter to check that we are getting a stable 5 volt reading from our microcontroller pin. 4. We will write in our lab notebook about if the ultrasonic sensor is giving valid readings with the given voltage input. We will also discuss if we had any odd readings from our voltmeter.





g. Motor Subsystem

Motor Subsystem:

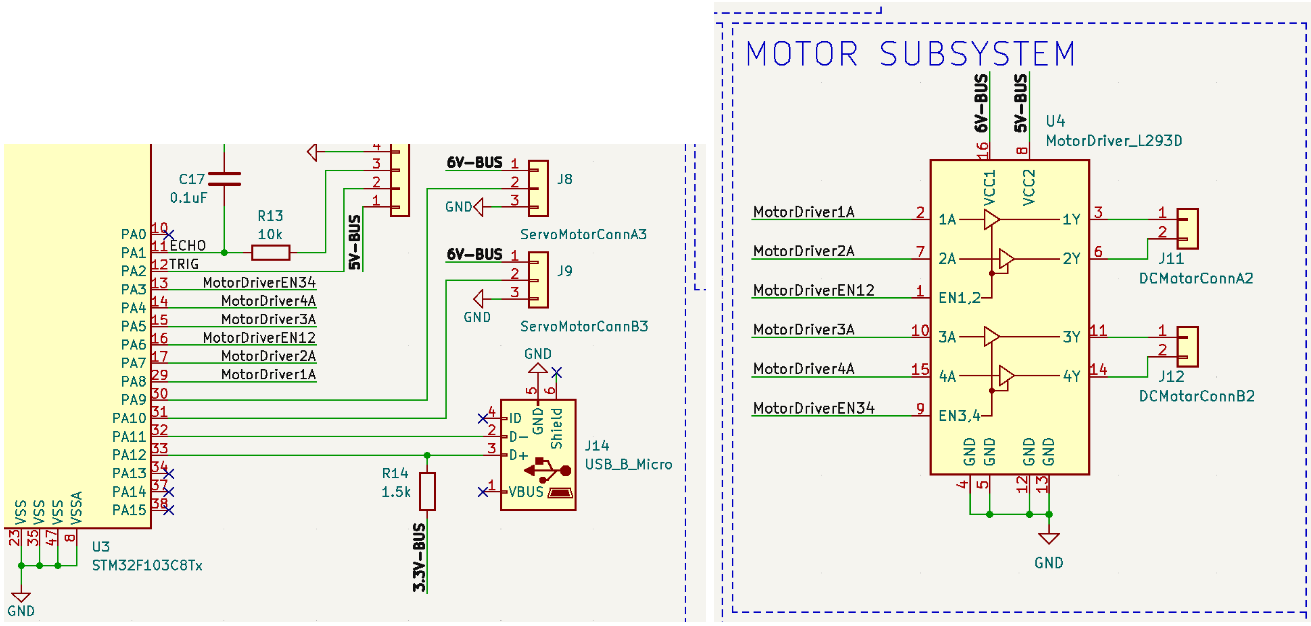
The purpose of this subsystem is to control the motors for the various functions that we need to perform. There will be 2 DC motors (6V) that will run with a constant speed to handle the shuffling mechanism, 1 servo motor (6V) that will be used to distribute the cards by launching them out of our device, and 1 servo motor (6V) to rotate the device the correct amount when dealing to each player. The subsystem will contain an H-bridge motor driver (L293D from STelectronics) to control the speed and direction of the 2 shuffling DC motors on our device through the PWM signals it receives from the microcontroller. These DC motors will be powered through the 6V regulated voltage outputting from the power system. The servo motors that are responsible for rotating the device and shooting out the cards when dealing will also be powered by the voltage regulated 6V output from the power subsystem but will not go through an H-bridge and will be directly controlled through a PWM pin on the microcontroller.

Requirements	**Verification**
Must be able to deal the cards and rotate our entire device with an operating voltage between 4.8-6.0V	1. Our dealing and rotation servo motors must be able to operate on 6V. Our 9V battery will be fed into an LM7806 (voltage regulator) to provide a stable 6V source. 2. We will validate that this operating voltage is sufficient by connecting our battery to

	the voltage regulator and directly feeding the 6V output to the VCC pin on the servo motor (one at a time) with a resistor in series. For a servo motor to operate it must be provided with a PWM signal, so we will also feed its PWM pin a 100% duty cycle signal. If the motors spin, this ensures that a stable 6V is sufficient for the DC motors to operate. 3. We will also validate that the power is correct and reliable by using a voltmeter to check that we are getting a stable 6 volts at the output of the voltage regulator. 4. We will write in our lab notebook about if the DC motors are giving valid readings with the given voltage input. We will also discuss if we had any odd readings from our voltmeter.
Rotation servo motor must have enough torque (>60 oz-in) to rotate the device	1. To verify that the rotation servo motor can provide >60 oz-in of torque, we will power the servo motor through the 6V regulated output from the power subsystem and the PWM signal from our microcontroller. We will send a signal to the servo motor to rotate 180 degree while applying a torque wrench to its shaft (known torque). 2. We will measure the angle of rotation of the shaft and record the time it spent rotating. 3. We will then calculate teh torque using the measured angle of rotation and the recorded time. We will then calculate the angular velocity and the actual torque delivered by the servo through the formula torque = load * distance / time 4. Lastly, we will compare the calculated torque to 60oz-in.
Must have the capability to rotate at least 180 degrees of freedom using a rotation servo in a circular and controller manner.	1. We have a servo motor that is able to rotate 360 degrees, however from a design standpoint, it will be hard to get 360 degrees of rotation due to wire tanglement. So we will be going for 180 degrees of rotation. Therefore it can be placed on the corner of a table. 2. This can be verified by checking manually that the servo motor is able to turn 180 degrees. It can also be checked by

	programming the motor and verifying that it can turn 180 degrees. 3. In our lab notebook, we will write if the servo is able to rotate this much, and we will also write about possibilities of making the design rotate more based on how tangled the wires are. We will also discuss the code that we write to program the motor. 4. The rotating servo motor shall stop at various angular positions based on the input of the user: number of players and game mode. This can be verified by inputting a specific angular position and checking the behavior of the servo.
Dealing servo motor needs at least 55.54 oz-in of torque in order to successfully launch a typical bicycle card (weighs roughly 100-150g)	1. The ability to shoot a deck of cards (one card at a time) can be verified by placing a deck of cards into the card bin, and checking if the card dealer deals cards one at a time. 2. The motion of separating one card from the deck at a time, and shooting the card out of the system onto a table can be verified by the following: defining the time it takes to retrieve one card from the bottom of the deck, and the torque necessary to shoot the retrieved card out of the system. The test case will be passed if exactly one card comes out. The test case will be a failure if zero OR multiple cards are trajected out of the system. 3. Trajecting the card towards a player in a way that the card is not flipped over upon landing will require fine tuning. Trajecting a single card towards a player given that the card will land within 12 inches in radial distance from the card dealer will also require fine tuning for torque.
Shuffling DC motors must be able to shuffle cards with an operating voltage of 6V.	1. Our shuffling DC motors must be able to operate on 6V. Our 9V battery will be fed into an LM7806 (voltage regulator) to provide a stable 6V source. 2. We will validate that this operating voltage is sufficient by connecting our battery to the voltage regulator and directly feeding the 6V output to each DC Motor (one at a time) with a resistor in series. If the motors spin, this ensures that a stable 6V is

	sufficient for the DC motors to operate. 3. We will also validate that the power is correct and reliable by using a voltmeter to check that we are getting a stable 6 volts at the output of the voltage regulator. 4. We will write in our lab notebook about if the DC motors are giving valid readings with the given voltage input. We will also discuss if we had any odd readings from our voltmeter.
Shuffling DC motor again must be able to output at least 40 oz-in of torque in order to shuffle the cards	1. To verify that the two DC motors of the card shuffler can provide >40 oz-in of torque, we will power the motor through the stable 6V supply from the power subsystem and control it through a microcontroller that feeds to our H-bridge motor driver. 2. We will apply a load through a torque wrench on the DC motor shaft and measure the current being drawn by the DC motor using a multimeter. 3. We will then vary the PWM signal being applied to eventually a 100% duty cycle such that the motor is applying its maximum torque. 4. We will then calculate the torque delivered by the DC motor through the following formula: torque = current*voltage/m 5. Lastly, we will compare the calculated torque to 60oz-in.



h. Tolerance Analysis

Identify an aspect of your design that poses a risk to successful completion of the project. Demonstrate the feasibility of this component through mathematical analysis or simulation.

The most difficult part of this project will be distributing the cards in such a way that they go the right distance and angle to the player without the cards being flipped or revealed. Error in this distribution of the cards can occur for various reasons including false measurements from the ultrasonic sensor, friction from the device platform, air resistance while the cards are traveling, the dealing motor not having enough torque/power to distribute the cards a long distance, and the slit not being the correct length to which it will only allow 1 card to go through it. Now to show the feasibility of all of the potential problems we might have:

Risks:

1. False measurements from the ultrasonic sensor
    - In order to combat this problem, we plan to mount the ultrasonic sensor in a straightforward direction so that every player can be detected by the sensor.
    - We also will do heavy testing to make sure that our ultrasonic sensor is outputting valid results by testing repeatedly with objects to determine their distance.
    - We will also do a simulation from objects of various lengths away to determine the maximum distance that our ultrasonic sensor is able to deal with accurately.

2. Friction from the device platform
    - In order to maintain the integrity of the card deck that is being used, the motors used to traject one card at a time shall be tuned to ensure that the friction between the cards and the device are not harmful to any given card in the deck.
3. Air resistance while cards are traveling through the air
    - To address this, we will simulate launches at various predicted distances using the following equation: $d = v_0\sqrt{\frac{2h}{g}}$ where v_0 is the initial velocity, h is initial height of the card, and g is the force of gravity. Since v_0 is dependent on the torque of the motor, we will use the following equation to calculate $v_0 = \frac{2\tau r}{m(R)}$: where τ is the applied torque, r is the radius of the wheel, m is the weight of the playing card, and R is the radial distance between the center of the wheel and the point of launch for the card (r+h, where h is the thickness of a playing card, typically 0.17-0.24mm). When launching at the various predicted distances, we will measure the error and account for it in future launches through developing a lookup table with offsets for different distance ranges.
4. The dealing motor not having enough torque/power to distribute the cards a long distance
    - We will initially test our servo motor to see if it provides enough torque to push the card and a certain specified distance. However, if that does not work as planned, we have a backup plan in mind, in which we will have the servo motor simply push a card one at a time through the slit to another small platform with a DC motor powered through a PWM pin on our microcontroller that launches the card immediately.
5. The slit not being the correct length to which it will only allow a single card through
    - The dimensions of a card are 64 mm by 89 mm by 0.17 mm to 0.24 mm thick. Using this knowledge, we know that the actual slit has to be wider than the 64 mm measurement, but it doesn't matter a whole lot what the width of the slit is because as long as it's over ~67 mm the card will fit through it. As for the height, this is the more important measurement, we want to make sure only a single card can go through at a time. Since we know that the size of a card can be anywhere from 0.17 mm to 0.24 mm thick, we need the height of the slit to be approximately 0.27 mm, this will give enough room for the card to squeeze through. At the same time, we would need a minimum of 0.34 mm to fit two cards through, so 0.27 mm is a good height to choose.

3. Cost and Schedule

a. Cost Analysis

Below is a list of parts that are required for the making of one autonomous card dealer.

Item	Part Number	Price	# of units	Total	Status
Microcontroller and Connectors to Subsystems					
~~STM32 ARM Microcontroller~~	~~STM32F103C8T6~~	~~25.99~~	~~1~~	~~25.99~~	
Microcontroller	STM32F103C8T6	$6.66	1	$6.66	
0.1uF (100nF) Capacitor	GRM155R71H104ME14J	$0.10	1	$0.10	
10uF Capacitor	GRM155R60J106ME05D	$0.10	1	$0.10	
1uF Capacitor	CL10A105KA8NNNC	$0.10	1	$0.10	
10k Resistor	RT1206FRE0710KL	$0.14	2	$0.28	
1uF Capacitor	GRM155R61E105KE11J	$0.10	4	$0.40	
Switch: SW SPDT	RF1-1A-DC-2-R-1	$1.26	1	$1.26	
SWITCH SLIDE SPDT 200MA 30V	EG1218	$0.68	1	$0.68	
Crystal 16MHz - for clock	ECS-160-10-36Q-ES-TR	$1.05	1	$1.05	
10pF Capacitor - for clock	GRM0335C1E100JA01J	$0.10	2	$0.20	
1.5k Resistor - for programmer	RT0603FRE071K5L	$0.10	1	$0.10	
USB-B Micro Connector	10118192-0002LF	$0.43	1	$0.43	
2k Resistor - for I2C display	RT0603FRE072KL	$0.10	2	$0.20	
0.1uF Capacitor - for UltraSonicSensor	CL05A104KA5NNNC	$0.10	1	$0.10	
10k Resistor - for UltraSonicSensor	RT1206FRE0710KL	$0.14	1	$0.14	
NRST Button Connector - 2 pin	DF13-2P-1.25DSA(05)	$0.30	1	$0.30	
7SegmentDisplayConnector - I2C disp - 4 pin	B4B-XH-A(LF)(SN)	$0.22	1	$0.22	
UltraSonicConnector - UltraSonic - 4 pin	B4B-XH-A(LF)(SN)	$0.22	1	$0.22	
ServoMotorConnector - Servo - 3 pin	B3B-XH-A(LF)(SN)	$0.20	2	$0.40	
Power Subsystem					
3x Adjustable Voltage Regulator	LM350 - T0220	$2.97	3	$8.91	
9V battery - for power	6LR61GCP	$1.20	1	$1.20	
Fuse (2.5Amps)	TR-1245UMFF2-5-R	$1.31	6	$7.86	
FUSE 2.5A 63VAC/DC SLOW 1206	0685T2500-01	$0.28	6	$1.68	
0.1 uF Capacitor	CL05A104KA5NNNC	$0.10	3	$0.30	

1.0 uF Polarized Capacitor	CL10A105KA8NNNC	$0.10	3	$0.30	
240 Resistor - from 9 to 6 V	RC0402JR-07240RL	$0.10	1	$0.10	
910 Resistor- from 9 to 6 V	RC0402JR-07910RL	$0.10	1	$0.10	
240 Resistor - from 9 to 5 V	RC0402JR-07240RL	$0.10	1	$0.10	
680 Resistor - from 9 to 5 V	RC0402JR-07680RL	$0.10	1	$0.10	
240 Resistor - from 6 to 3.3V	RC0402JR-07240RL	$0.10	1	$0.10	
390 Resistor - from 6 to 3.3V	RC0402FR-13390RL	$0.10	1	$0.10	
Motor Subsystem					
~~L293D Motor Driver IC~~	~~SLRS008B~~	~~3.083~~	~~1~~	~~3.083~~	
Motor Driver IC	296-9518-5-ND	$4.29	1	$4.29	
Card Shuffler	Rareidel Company	$15.99	1	15.99	
Servo	ROB-11965	$13.95	2	27.90	
DC Motor Connector - 2 pin	DF13-2P-1.25DSA(05)	$0.30	2	$0.60	
Sensing Subsystem					
~~Ultrasonic Distance Sensor - HC-SR04~~	~~SEN-15569~~	~~4.50~~	~~1~~	~~4.50~~	
UltraSonic Sensor	3647-HC-SR04-ND	$6.55	1	$6.55	
User Interface					
7 segment display	COM-11441	14.95	1	14.95	
10k Resistor - for UI	RT1206FRE0710KL	$0.14	4	0.56	
1uF Capacitor - for UI	CL10A105KA8NNNC	$0.10	4	0.40	
Button	SW-PB1-1DZ-A-P1-A	$0.32	4	1.38	
Button Connector - UI - 2 pin	DF13-2P-1.25DSA(05)	$0.30	4	1.20	
			Total Cost of Materials	**$106.40**	

b. Estimated Hours of Development and Overhead Costs

The average pay of a new grad in ECE from UIUC in 2021 is $105,352, an hourly rate of $54.76. If we would account for the work time spent on the project per week from each team member, the cost of production would be a theoretical $\frac{3\ members}{} \cdot \frac{12\ hours}{member} \cdot \frac{\$54.76}{hour} \cdot 8\ weeks\ = \$15,770.88$.

Category	Adam	Rohit	Ralph

Circuit Design	35	20	20
Mechanical Design	20	35	20
Microcontroller Dev and Autonomy	20	20	35
Board Layout and Components Check	10	10	10
Soldering	5	5	5
Prototyping and Debugging/Testing	60	60	60
Documentation and Logistics	40	40	40
Total Cost of Production: $54.76**(Hourly Rate)** ∗ 200*3**(Total Estimated Hours)** = $15,770.88			

The grand total cost for this project is a projected including the cost of production and the cost of materials is the following: $\$15,770.88 + \$93.453 = \$15,864.333$.

c. External Materials and Resources

- **Machine Shop**
  Given that this project consists of many mechanical components, it is important to consult the ECE Machine Shop for guidance and assistance in creating a viable enclosure to hold the electronics and a durable system. We have consistently spoken to the Machine Shop after our weekly TA meeting on Wednesday's to ensure our progress is sustained.
- **Senior Design Lab Resources**
  Our team plans on utilizing the ECE's Senior Design Lab as a testing ground for the autonomous card dealer. Namely, we plan on using the soldering stations, computer stations, oscilloscopes, and power system supplies to test and showcase progress.
- **Development Resources**
  In order to create an autonomous card dealer, we will be programming on the STM32 ARM Microcontroller, in which we will learn to develop and test how the Microcontroller interfaces with the various subsystems involved in the project. In particular, we will be looking at resources to help us convert Sensor and Button Signals into useful data, and control the autonomous card shuffler and dealer's respective motors in a scheduled manner.

d. Schedule

Below is a projected schedule of our goals and several hard deadlines for the ECE 445 course. We plan to work on all of the parts of the project together, splitting up larger tasks when necessary based on interest.

Week	Week of	Goals
0		Design Document, PCB/Schematic Review, Order Parts
1	02/27 - 03/05	Order PCB first prototype, Board Layout Review
2	03/06 - 03/16	Mechanical Parts Assembly, Machine Shop
3	03/13 - 03/19	Spring Break, Microcontroller Development
4	03/20 - 03/26	Microcontroller, Sensor Development and Motor Testing
5	03/27 - 04/02	Second Round PCB Orders, Individual Progress Reports
6	04/03 - 04/09	MVP, basic functionality implemented (shuffling and dealing)
7	04/10 - 04/16	Team Contract Fulfillment
8	04/17 - 04/23	Mock Demo, Debug, Review parts and for Final Demo, Final Tune Up
	04/24 - 04/31	Final Demo, Final Review of Presentation and Paper
9	5/1 ++	Final Presentation, Final Paper

Week	Week of	Adam’s Goals	Ralph’s Goals	Rohit’s Goals
0		Design Document, PCB/Schematic Review, Order Parts	Design Document, PCB/Schematic Review, Order Parts	Design Document, PCB/Schematic Review, Order Parts
1	02/27 - 03/05	Order PCB first prototype, Board Layout Review	Order PCB first prototype, Board Layout Review	Order PCB first prototype, Board Layout Review
2	03/06 - 03/16	Mechanical Parts Assembly, Machine Shop	Mechanical Parts Assembly, Machine Shop	Mechanical Parts Assembly, Machine Shop
3	03/13 - 03/19	Spring Break	Spring Break	Spring Break

4	03/20 - 03/26	Microcontroller Development and Motor Subsystem Testing	Microcontroller Development and Power Subsystem Testing	Microcontroller Development and Sensing Subsystem Testing
5	03/27 - 04/02	PCB Edits, Second Round PCB Orders, Individual Progress Reports	PCB Edits, Second Round PCB Orders, Individual Progress Reports	PCB Edits, Second Round PCB Orders, Individual Progress Reports
6	04/03 - 04/09	MVP, basic functionality implemented (work specifically on shuffling), Last PCB Edits and Orders	MVP, basic functionality implemented (work specifically on dealing), Last PCB Edits and Orders	MVP, basic functionality implemented (work specifically on UI), Last PCB Edits and Orders
7	04/10 - 04/16	Link all components of the project together, Add additional functionality, Team Contract Fulfillment	Link all components of the project together, Add additional functionality, Team Contract Fulfillment	Link all components of the project together, Add additional functionality, Team Contract Fulfillment
8	04/17 - 04/23	Mock Demo, Debug Motor Subsystem and shuffling functionality in specific	Mock Demo, Debug Power Subsystem and dealing functionality in specific	Mock Demo, Debug Sensing Subsystem and UI functionality in specific
	04/24 - 04/31	Debug any last minute things as a group, Final Demo, Final Review of Presentation and Paper	Debug any last minute things as a group, Final Demo, Final Review of Presentation and Paper	Debug any last minute things as a group, Final Demo, Final Review of Presentation and Paper
9	5/1 ++	Final Presentation, Final Paper	Final Presentation, Final Paper	Final Presentation, Final Paper

4. Discussions of Ethics and Safety

In terms of the ethics of this project, we mainly chose to follow the IEEE Code of Ethics [1]. The main purpose of this project is to prevent cheating and make card games more fair by eliminating the role of a dealer. It makes the game fair by allowing the machine to take care of anything outside of actually playing the game.

1. *IEEE Code of Ethics:* To uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities.

   a. To ensure the health, safety, and welfare of anyone who may use the card dealer we are developing, we will make sure to thoroughly test each of our components for proper functionality.
   b. As our project is quite mechanical, we will also frequently meet the Machine Shop Technicians to make sure none of the mechanical components or our design can prove hazardous to a user, or if any uncertainties surface during the development of our project.
   c. Our team will also meet with expertise (Teaching Assistants, Professors, etc.) and accept any constructive criticisms of our project.
2. *IEEE Code of Ethics:* To treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others.
   a. To ensure proper teamwork etiquette, our team shall practice frequent and efficient communication via the following method: Discord servers with TA (Nikhil Arora) and text message group chats. Documentation and designs related to the project SHALL be shared virtually on a Google Drive to ensure ease of use and accessibility and physically on lab notebooks. This allows our team to keep track of the whereabouts of the project and have a running log of the progress the team has made.
3. *IEEE Code of Ethics:* To strive to ensure this code is upheld by colleagues and co-workers.
   a. To ensure this code is upheld by colleagues and co-workers, our group has decided to schedule a weekly meeting to discuss any violations of the code of ethics that may surface throughout the semester. If any matters are brought up, we will discuss them with our TA and/or the professor.

In regards to the safety and regulations in this project:

1. For each of the motors used to rotate the base, shuffle the deck, and traject a given card, the housing for the cards and wire harness SHALL be protected from the following events: (1) wire twists and entanglements (2) cards ‘accidentally’ falling in between 2 moving parts. Given that the project is mechanical movement heavy, it is important to consider (1) ease of use (2) limiting system failure due to moving and rotating parts (3) noninvasive to any given card deck that is to be operated on. The system shall have a kill switch (ON-OFF) button to ensure that all power is cut when prompted.
2. When launching the cards in the players’ directions, it is important to send a ‘safe’ amount of power that ensures no-injury to the player. Too little power, and the dealer system will be useless. Too much power, and the dealer system may cause injury to the users. The direction and speed of

the card trajectory SHOULD be consistent and predictable in order to minimize the chances of injury to the user(s).

3. Given that the device holds electronic components, it is important to ensure that the enclosures and housings can keep the electronic hardware safe. At the minimum, our device's electronic housings will protect the electronics from liquid spillage and wire entanglement.
4. Since ultrasonic sensors produce such high frequency signals, having too much exposure to these noises can damage your hearing [2].We shall set a limit on the use of the ultrasonic sensor by only turning it on when the device is in "DEALER MODE ONE"

5. Citations

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

[2] Moyano, David Baeza, et al. "Possible Effects on Health of Ultrasound Exposure, Risk Factors in the Work Environment and Occupational Safety Review." *Healthcare (Basel, Switzerland)*, U.S. National Library of Medicine, 24 Feb. 2022, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8954895/.