No Chips? No Problem. - Poker 2.0

By

TJ Lambert

Nathaniel Stoll

Zach Wong

Final Project Report for ECE 445, Senior Design, Spring 2020

TA: Stephanie Jaster

8 May 2020

Project No. 4

Based off of Project No. 62 from Spring 2019, “Electronic Betting System”

Abstract

Card games, specifically poker, have risen in popularity amongst people due to the simplicity of materials needed to play combined with the potential to win money. While many play the common variant, Texas Hold'em, most find themselves unable to play leisurely on their own schedule due to the lack of physical poker chips readily available. Team 62's project, "Electronic Betting System For Poker," introduced a system where the entire game of poker can be played electronically. However, the product's limitation on the number of players capable of playing mixed with an unreliable communication protocol stemming from the main console gave motivation for our solution. In our design, we not only removed the player limitations as well as implemented a more reliable communication protocol, we also made the design more versatile by allowing people with physical poker chips to use our design as well.

Contents

1 Second Project Motivation

1.1 Updated Problem Statement

Poker is not only the most popular card game in the world, but it is also the most commonly referenced card game [1]. It has slowly evolved from being a physical, in-person card game to being played online with players facing off from around the globe. Las Vegas, one of the world's biggest casino epicenters, earned up to $8 million in revenue from poker rooms in a single month [3]. While the popularity and demand for poker is undeniable, it unfortunately is a game that cannot be played within the comfort of one's home unless they have poker chips. The plastic coins are an integral part of the game as it encompasses the gambling portion of poker, which entices players with the possibility of winning money. However, not everyone has these plastic coins already in his/her possession, and those who do, may not have enough to go around based on the number of people wanting to play. This problem causes people to rely on going to businesses and corporations that supply the necessary materials. A similar problem was proposed by Team 62 in Spring 2019 with their "Electronic Betting System for Poker" [2]. While their design simulates the entire game of poker electronically, it falls victim to limiting the number of players capable of playing based on the number of RFID enabled remotes. Additionally, the design implements an unsteady, unreliable communication protocol between the main console and remotes causing slow gameplay. We not only want our design to improve on Team 62's design flaws, but also solve the problem of making the game of poker more practical for average consumers without having to rely on big businesses or corporations. We believe our proposed problem is worth solving as the game of poker is undeniably popular around the world, and we believe every person should be given the chance to play with as few limitations as possible. From the data previously mentioned, we can safely say there is an audience of people who would be interested in the solution to this problem. Additionally, a couple of team members have had first-hand experience playing poker themselves as well as have had friends/colleagues who play too. Therefore, not only do we see this problem occuring in public society, we also have a personal connection to it.

1.2 Updated Solution

The goal of our project is to develop and implement a solution that is able to simulate a game of poker without limiting the number of players wanting to play and have the game be playable with people who have poker chips as well as those without poker chips. Additionally, we want to maintain the key components of poker-like face-to-face interaction as it used to tell if someone may be bluffing or not. For our solution design, we plan to create a main console unit. The main console unit will have an insert that allows players to insert poker chips into it if available. When inserted, the poker chips will be moved into a rotated drum run by servo motors that push each chip down a funnel. Once in the funnel, an additional servo will stop the chip from falling and have an IR LED/sensor as well as a color sensor to detect the presence and color of a poker chip. Once detected, the servo will allow the chip to fall through into a pull-out tub that the winner of each round will use to collect the pot. Additionally, an LCD display will be mounted onto the control unit to display the current pot as well as the current amount of each player. The game will be updated and controlled through the use of an internal microcontroller that will receive sensor

data as well as data coming from external devices. Each player will use their own external device (laptop, smartphone) and open an internet browser to connect to the main console's IP address. Once connected, an internal Wi-Fi module will regulate HTTP requests sent from the microcontroller to the devices. Through this browser interface, players will have the ability to check, raise, and fold with their actions being reflected on the LCD display. Both the main console unit and external devices will ensure our project solution adequately solves the proposed problem.

Our design solution solves the previously stated problem by creating a design that can simulate a game of poker electronically as well as when physical poker chips are present. Additionally, the compact design our solution provides makes it practical for average consumers to use in their households and use at their leisure. Our design solution also provides a more versatile and reliable product than Team 62's solution. Wi-Fi connectivity is much more reliable than using RFID as RFID requires more readers to provide better accuracy [4]. More readers result in higher costs for the RFID infrastructure too. Additionally, RFID can easily be interfered with by other RF signals as well as Wi-Fi access points and since most households have a Wi-Fi system in place it seems redundant to have a product require a completely different system to be installed. Within their final report, Team 62 did note how the combination of RFID and RF signals was easily interferrable and unreliable suggesting in future designs a more reliable protocol should be implemented [2]. Furthermore, their solution also incorporates remotes that help dictate whose turn it is, which limits the number of players to the number of remotes available. Using an internet browser interface that can be connectable through mobile devices and/or laptops removes that limitation as players will only need to have a smartphone which, in today's society, most people already have.

On the market, there are a few existing solutions that attempt to solve our problem, but none that directly eliminate the problem entirely. One of these solutions that is commonly found is simulated poker online through mobile applications. There are a ton of online poker applications scattered throughout the app store. However, most of these online poker mobile applications remove the element of face-to-face interaction, which is a key component many players use to tell if other players are bluffing or not. There is one mobile application called Bold Poker that still gives players face-to-face interaction. Bold Poker simulates the table, dealer, cards, and poker chips on an iPad with players playing together through their mobile devices [4]. While this product does eliminate the issue of limiting the number of people wanting to play, it does not take into account those consumers who have poker chips already in their possession. Additionally, Bold Poker requires a 5 minute delay every 30 minutes calling it a "cigarette break," which can be very annoying and cumbersome [5]. To remove this "cigarette break," players will have to pay $4.99 to avoid any interruptions [5]. Bold Poker also requires players to have an iPad to simulate the poker table. No Android or other tablet can be used. This requirement severely limits the audience scope of this product as an iPad is not cheap and has a starting price tag of $300 with the potential to reach $1500. Other solutions aimed to solve our design problem consist of building entire apparatuses to mimic a casino poker table or electronic slot machine. It should be noted that both these designs are currently patented, but have yet to reach the public as an eligible product. The casino poker table suggests the solution should consist of a full embodiment of a poker table with an electronic dealing system [6]. This solution still contains a limit on the number of players able to play with only six seats available. Additionally, this solution is only feasible for businesses and corporations who have the space and money

to purchase such a product. The electronic slot machine has the same issues. The size of the machine makes it only feasible to business and corporations [7]. This slot machine also does not take into account physical poker chips, but rather makes everything electronic [7]. While there are solutions existing in the market or making their way to it, none of them provide direct solutions to the problem we are trying to directly solve. Problems like setting a limit on the numbers of players and giving players the option of still playing poker despite not having physical poker chips are just some of the issues that still linger today. Our project solution aims to tackle and solve each of those issues as well as make it feasible to everyone.

1.3 Updated High-Level Requirements List

- The entire system, main unit and website, correctly performs all functions of a poker game, specifically Texas Hold'em, without the use of physical poker chips.
- The poker chip detector must have greater than 95% accuracy in correctly determining the quantity and color (blue, black, green, red, or white) of chips inserted into the console during a person's turn.
- The console must be able to successfully handle requests from external devices for the "game dashboard" (an HTML file displaying players' money totals, whose turn it is, and options to buy in, check, call, raise, and fold) 90% of the time.

1.4 Updated Visual Aid

Figure 1 shows two diagrams of our console. The inside view displays internal components and measurements that a user would not normally see. The outside view shows what the product would look like when a player is using it.

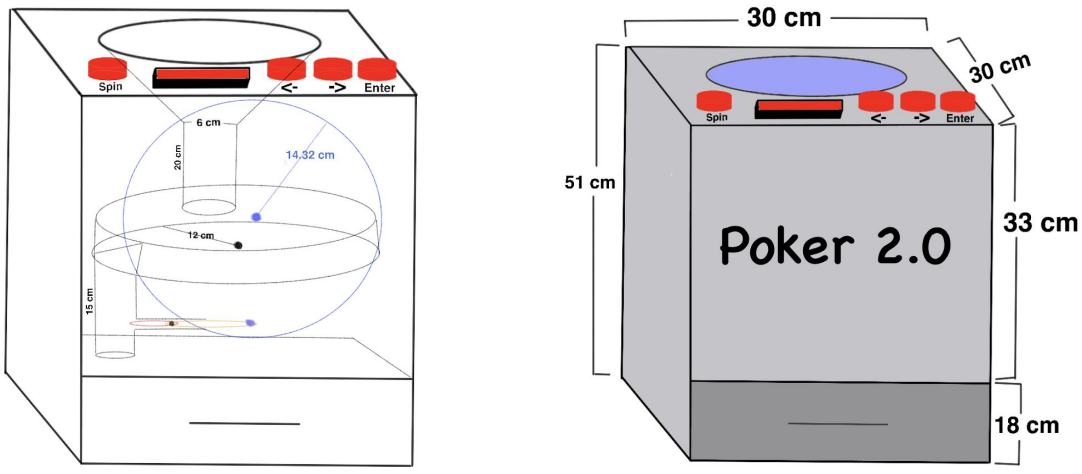


Figure 1. Inside view of console (left) and outside view of console (right).

1.5 Updated Block Diagram

Figure 2 shows how all of these modules physically connect with one another as well as how the console communicates with other devices on the wireless network (that is, through a router).

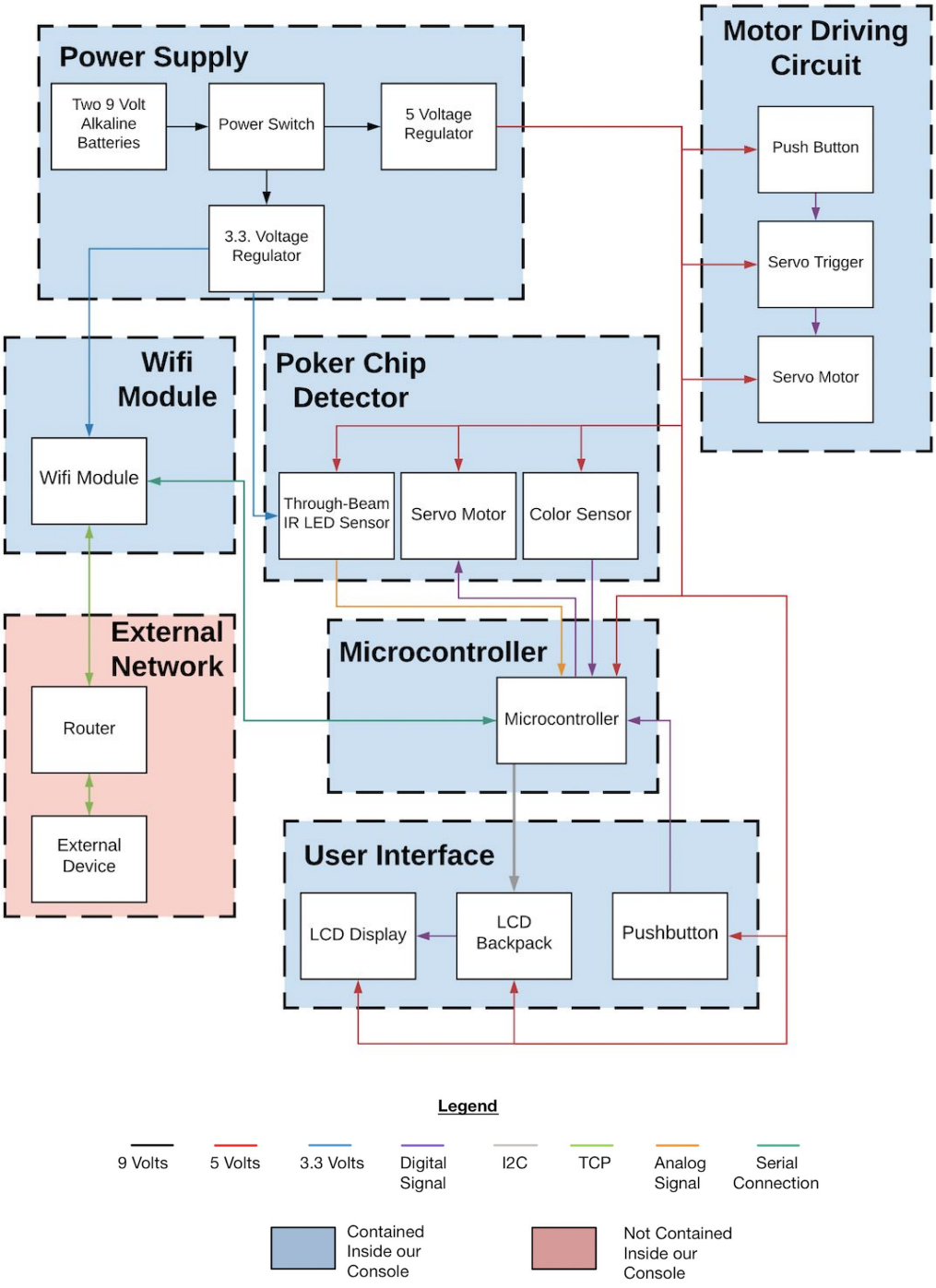


Figure 2. Block diagram.

2 Second Project Implementation

2.1 Implementation Details and Analysis

The overall functionality of the entire system depends largely on one machine: the main console. Inside this machine, there are six subsystems: a power supply circuit, a poker chip detector, a motor driving circuit, a microcontroller unit, a Wi-Fi module, and a user interface module. Sections 2.2-2.5 describe these submodules in greater detail and show their relevance to the high-level requirements of the project.

2.2 Power Supply

Every single component in Figure 2 needs some kind of electrical power to operate—without power, the machine does not work. In general, this project consists of two voltage levels: 5 V and 3.3 V. To accommodate for this, the power supply uses both a 5 V and a 3.3 V switching regulator, along with a rechargeable 7.4 V, 2.2 Ah battery and a power switch. Figure 3 contains the schematic for this module.

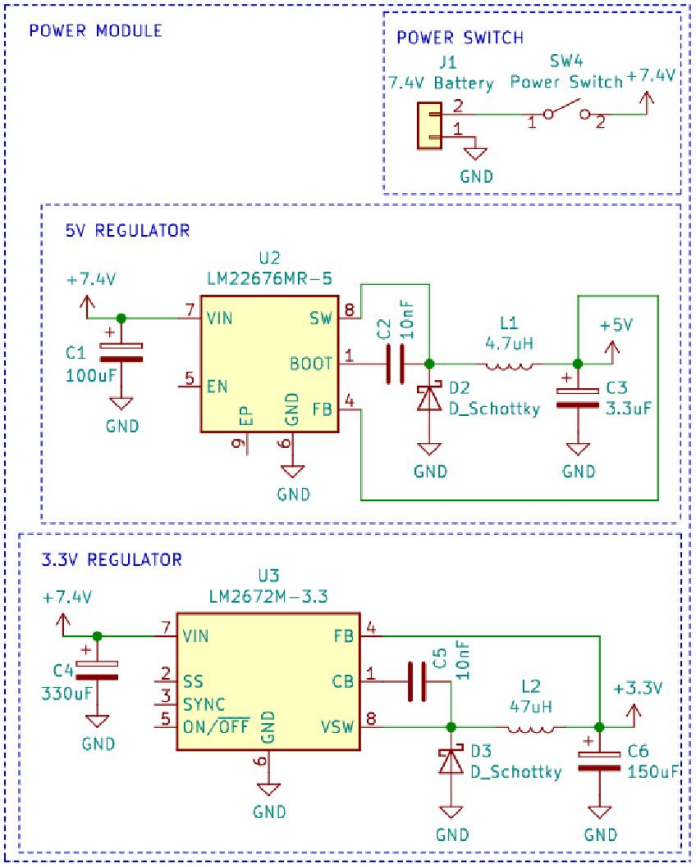


Figure 3. Power supply circuit schematic.

2.3 Poker Chip Detector

In order for people to use poker chips with the console, somehow, the console must keep track of the quantity of chips inserted into it as well as the color—that is, the value—of each chip. Ultimately, there are three pieces of hardware which indirectly work together to accomplish these two objectives: a through-beam IR LED sensor, a color sensor, and a servo. As players insert chips into the spinning drum, they fall one-by-one through the console. As each chip falls, the servo arm momentarily catches one chip at a time. Now, with the servo arm holding the poker chip steady, the through-beam IR LED sensor detects the presence of the chip and signals this to the microcontroller. Then, the microcontroller takes a reading from the color sensor and performs a cycle of the servo lever to let the chip drop into the bin as well as catch the next chip for the next reading. This process repeats until there are no longer any chips in the channel. Overall, the poker chip detector needs to provide accurate performance as it directly relates to the second high-level requirement: reliably scanning poker chips. Figure 4 shows the schematic for this module.

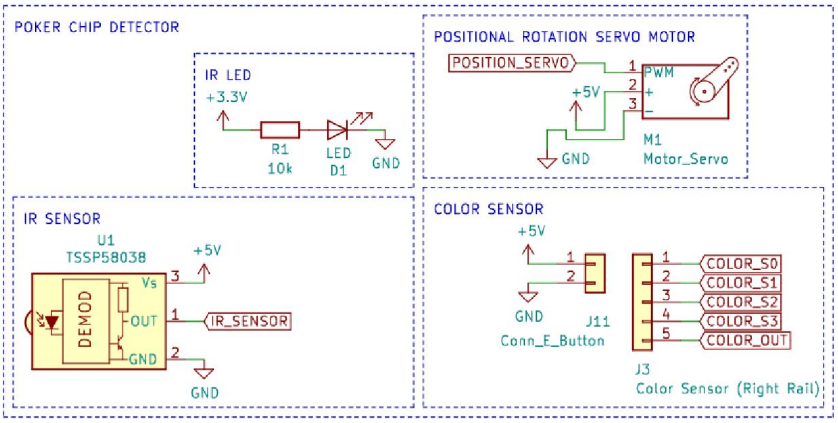


Figure 4. Poker chip detector circuit schematic.

2.4 Motor Driving Circuit

In order for players to use poker chips with the console (which, again, is one of the key differences from this solution compared to the previous team's solution), there must be some kind of mechanism in place to direct the chips through the machine. This is the purpose of the motor driving circuit. At the top of the console, players place their poker chips into a funnel which feeds them into a drum. This drum consists of a spinning disk located at the bottom which slides the chips into a slot inside the console. To electrically control the spinning of the disk, the console uses a servo motor to physically rotate the disk and a servo trigger to drive the servo. The user presses a pushbutton to communicate to the servo trigger to turn the servo on and off. Ultimately, this mechanism saves time during each person's turn (since players can feed multiple chips into the machine at once without having to place them in one by one), and it ensures chips move correctly through the machine so the poker chip reader does not run into issues when it tries scanning the chips. Figure 5 shows the schematic for this module.

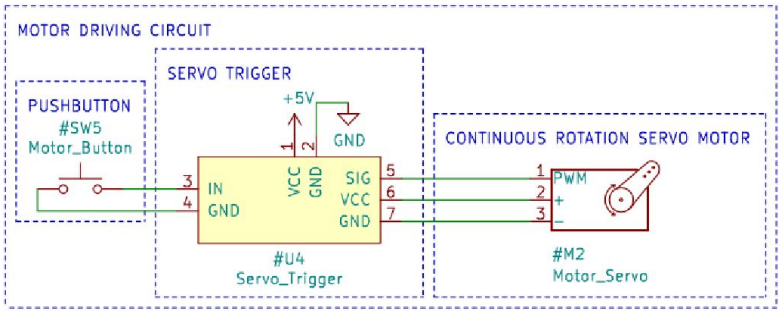


Figure 5. Motor driving circuit schematic.

2.5 Microcontroller

At the heart of the console lies the ATmega328P microcontroller. Nearly every subsystem (with the exception of the motor driving circuit and the external network) directly interacts with the microcontroller, and as such, it is responsible for many different things. First of all, it is the device which houses game state data. Next, it is responsible for reading input from the "select" button on the user interface and printing values and instructions on the LCD screen. Additionally, it interprets the quantity and color of poker chips read by the poker chip detection subsystem and pulses the position rotation servo motor to let chips fall through the console. And finally, given all of the information it stores about the game, it interacts with the Wi-Fi module which, in turn, interacts with the external network so people have access to this data via a browser on an external device. Because the microcontroller interacts with essentially all of the components in the console, it is an instrumental piece of the machine as a whole, and ultimately, impacts whether or not the first and third high-level requirements are met (since it controls gameplay and, in the end, is what users interact with). Figure 6 shows the schematic for this module.

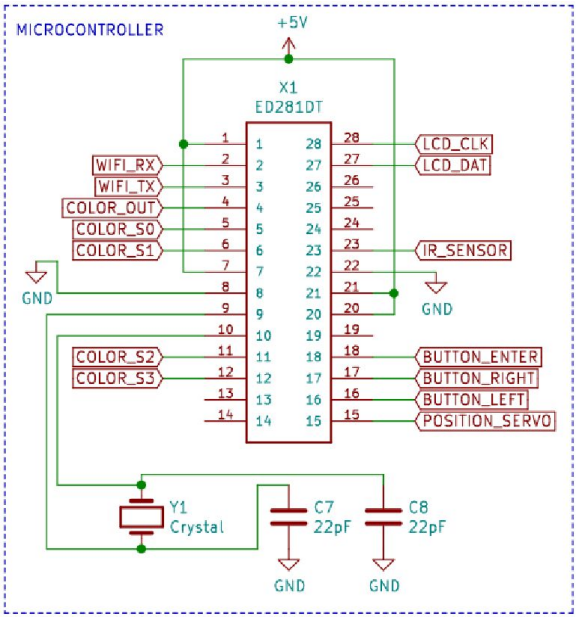


Figure 6. Microcontroller circuit schematic.

2.6 PCB Layout

Incorporating all of the components from the block diagram (excluding the motor driving circuit which works independently from everything else), the console's PCB has a width of 84.5 mm and a length of 94.5 mm. Figure 7 shows the layout for this PCB.

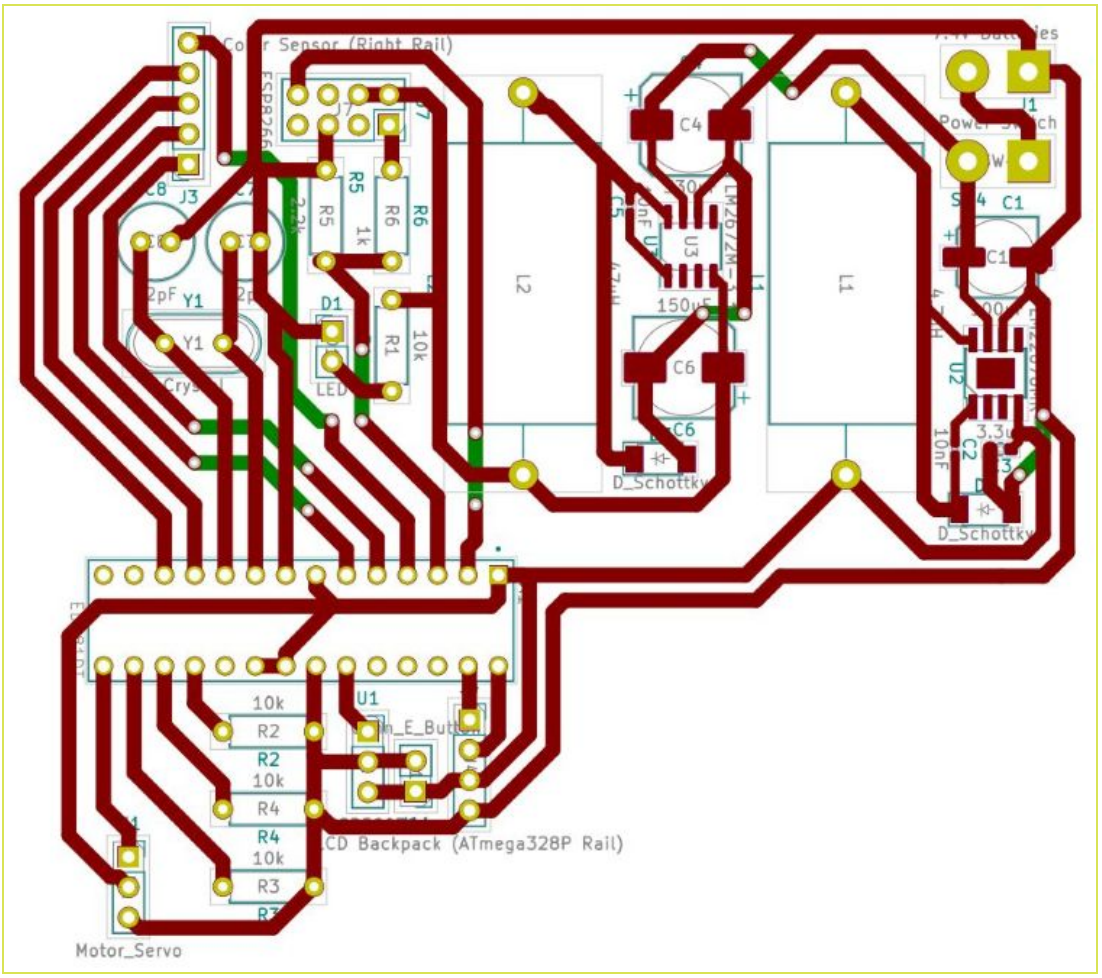


Figure 7. Console PCB layout.

2.7 Tolerance Analysis

One of the most important aspects of this project in terms of gameplay is the console accepting poker chips quickly and reading them accurately to give the players the correct credit. While we brainstormed many different ideas on how to accept the chips, one way became the clear winner for our purposes.

2.7.1 Rotating Disk

Some antique coin counters utilized a spinning disk. A user would place their coins on this disk and the centripetal force pushes the coins to the outside to be sent through a slot that can only fit one coin at a time [10]. We decided that this solution is better than a slot to insert each chip individually since, in poker, a player can go all in and would spend a very long time inserting each chip. With the spinning disk, a player can simply dump all their chips and not have to do anymore work. Some questions that arise with the spinning disk are how large should the disk be, how fast can the chips go through the collector, and how fast will it spin? The average poker chip's diameter is 39 mm but some chips are 40 mm [11]. We decided that a good size disk would be one that can fit a row to accept the chips and have enough room for other chips to go around the circle again. We also want to avoid having a larger than necessary disk so our overall design will be more portable. We selected a radius of 12 cm so that almost three chips can fit along the radius. We have selected the 1528-1496-ND Continuous Rotation Servo Motor, and from its datasheet, we know it has an average angular velocity of 1 rotation / 1.08 seconds or 0.9259 rotations / 1 second [16].

If our chips are lined back to back on this rotating disk, which is very possible they will be if someone adds a lot of chips, we want to know how fast these chips will be entering the exit slot so we can calculate if our sensors will have enough time to find the color and prevent blockage in our system. Figure 8 shows how we can identify the angle of the disk each chip will fill. We have decided to simplify this expression by treating the chips as a flat line with the length of the diameter of a chip. This will make the calculations easier and will also speed up the rate the chips will exit giving us a little leeway.

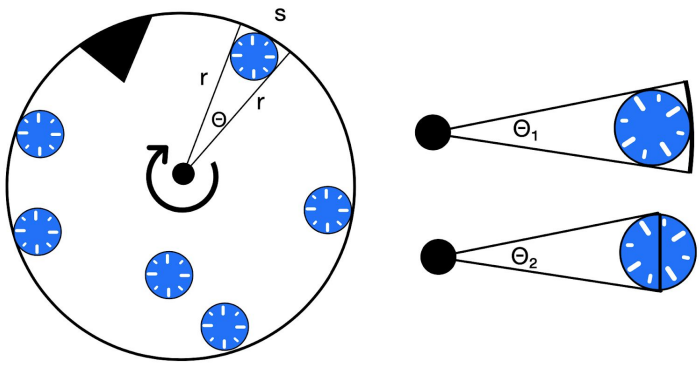


Figure 8. Diagram showing angle Θ that covers a poker chip. Rotating disk has radius r, and s is the arclength of that angle Θ. Diagram also shows simplification going from solving Θ_1 to Θ_2.

In order to find Θ, we can break this triangle up into two identical right triangles as shown in Figure 9. We can solve for *a* by using Pythagorean Theorem as shown in Equations (1)-(4).

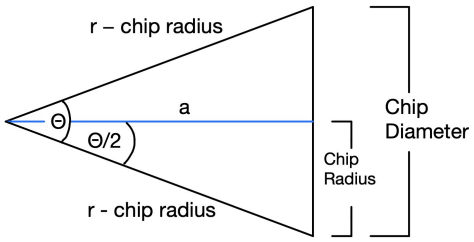


Figure 9. Isosceles triangle formed by the chip's radius, Θ, and the chip's diameter.

$$a^2 + 19.5^2 = (120 - 19.5)^2 \quad (1)$$

$$a^2 = 100.5 - 19.5^2 \quad (2)$$

$$a = \sqrt{(100.5^2 - 19.5^2)} \quad (3)$$

$$a = 98.59\ mm \quad (4)$$

Now we can use this r to solve for Θ/2 shown in Equations (5)-(6). We can take that Θ/2 and multiply by 2 to get Θ in Equation (7).

$$\Theta/2 = tan^{-1}((chip\ radius) / a) \quad (5)$$

$$\Theta/2 = tan^{-1}(19.5 / 98.59) = 11.19° \quad (6)$$

$$\Theta = 11.19° * 2 = 22.38° \quad (7)$$

We will now take this Θ and multiply it by the average angular velocity of the rotation disk shown in Equation (8) to find out how much time it takes for a chip to move one chip length on the edge of the disk.

$$(1.08\ s / 360°) * 22.38° = 0.06714\ s \quad (8)$$

This tells us the rotating disk may spit out a chip every 67.14 ms, and we need to deal with them quickly or the tube can possibly become clogged up.

2.7.2 Color Sensor

One of the most important parts of this system is the color sensor. The color sensor will scan each chip and give it value and without it, we would not be able to use poker chips. In order for the color sensor to translate the color, it needs to be able to see the chip for a minimum amount of time. The color sensor has four LEDs on it with different frames to detect intensities of red, green, and blue. The fourth LED is a clear frame, but we will not use that one. The datasheet for the color sensor has sample code that gives each LED a 10 ms timer for getting the values [12]. The color sensor is also required to do two

digitalWrites in order to turn a different LED and each digitalWrite takes about two ms [13]. Using Equation (9), we can see that the color sensor needs about 42 ms to test the three different color intensities.

$$(10\ ms + 2 * 2\ ms) * 3 = 42\ ms \tag{9}$$

The question now is where can this color sensor go in order to have enough time to read the values. The tangential velocity of the chip leaving our rotating disk is derived in Equations (10)- (11) by taking the center of mass distance of the poker chip on the outer wall of the rotating disk and multiplying it by the average angular velocity. In reality, the disk will most likely bounce around slowing down the tangential velocity, but we will need to be prepared for the case where it does not.

$$Center\ of\ Mass\ of\ Poker\ Chip\ Radius\ =\ 120\ mm\ -(\ 39\ mm / 2\ )\ =\ 100.5\ mm \tag{10}$$

$$V_{tan}\ =\ 0.1005\ m\ *\ 2\pi\ /\ 1.08\ s\ =\ 0.58468\ m/s \tag{11}$$

If we place the color sensor right where the chip exits the rotating disk, the chip will only be in front of the color sensor for a limited amount of time. The difference between the length of the color sensor and poker chip is 10.6 mm. Equation (12) shows us how much time the chip will spend in front of the color sensor.

$$t_1 - t_0\ =\ 10.6\ \ mm\ *\ (1\ m / 1000\ mm)\ *\ (1\ s / 0.58468\ m)\ =\ 18.13\ ms \tag{12}$$

Because the poker chip needs to be in front of the color sensor for at least 42 ms, we can see that having the sensor right at the exit of the rotating disk will not work.

2.7.3 Servo Motor

We need to stop the chips after they come out the rotating disk just long enough for the color sensor to read the values. After that we want to make sure we go to the next chip so that the tube does not get clogged up. In order to do this, we are going to utilize a servo to halt a chip, move to let that chip through, then stop the next chip. In Figure 10 we can see that the width of the tube for the chips to pass through is 42 mm so that there is enough room for both 39 mm chips and 40 mm chips to fit. Our servo will be connected to a disk that can push a lever in and out of the tube to control the traffic of chips.

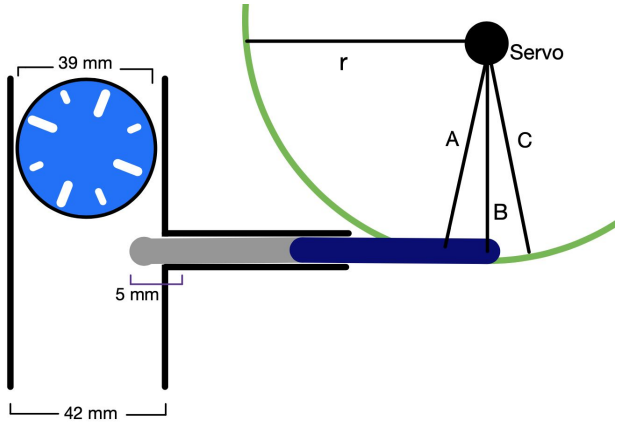


Figure 10. Diagram of chip size compared to the tube as well as the servo disk connected to a lever to control chip traffic in the tube.

Because our rotating disk can possibly spit out a new chip a little bit slower than every 56.1 ms, we need to rotate the chips as closely to the time it takes the color sensor to read the values as we can (42 ms). We want our lever to move about five mm to make sure there is enough room to block the chip and completely let it through as well. According to the Arduino tutorial for servo motors, it is good to give the servo motor 15 ms per degree [14]. In one cycle our servo will need to go out and then come back. If we move two degrees out and then two degrees back in, that will take 60 ms which is about all we will have time to do. Although two degrees does not seem very significant, we can make our disk that the servo is connected to large enough that two degrees can give us our requested lever distance. Figure 10 shows how our servo connected to our disk and lever will look. In the diagram, we have listed three different positions on our servo's disk labeled "A", "B", and "C". Equations (13)-(17) show how large the radius of this disk needs to be to move the lever five mm in two degrees. We also simplified these equations a little since the arclength of two degrees will almost be flat, we decided to treat the five mm we want to move as our arclength.

$$s = r\,\Theta \tag{13}$$

$$0.005\ m = r * 2° \tag{14}$$

$$0.005\ m = r * 2° * (2\,\pi / 360°) \tag{15}$$

$$r = (0.005\ m * 360°) / (2° * 2\,\pi) \tag{16}$$

$$r = 0.1432\ m \tag{17}$$

While this seems like it should be sufficient for the system, we still need to consider the speed at which the next chip will fall after we pull out this lever. Will the lever be able to go back in before the next chip falls through? Figure 11 shows the different stages the lever can take.

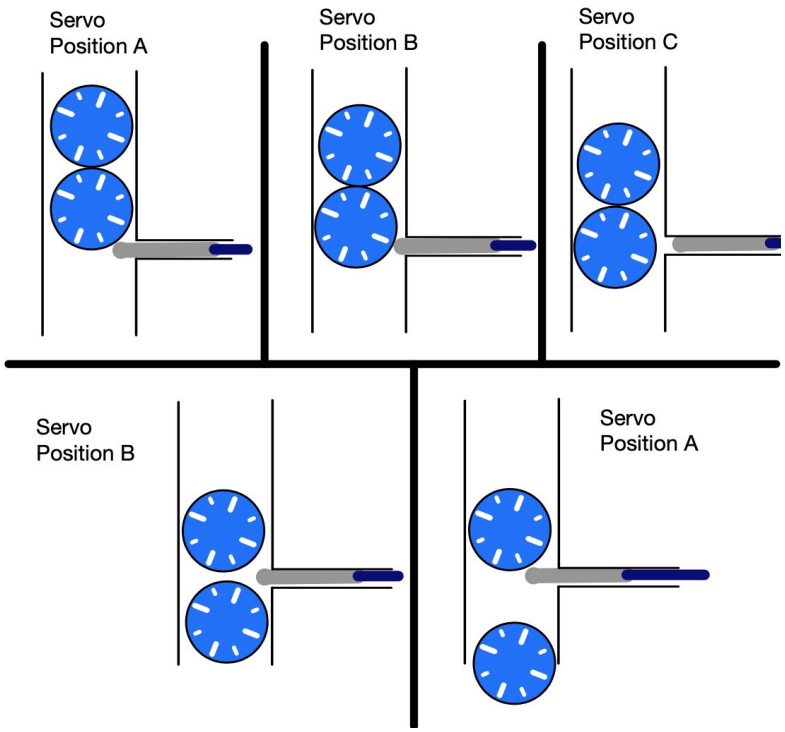


Figure 11. The lever positions based on the servo motor positions.

While the lever is moving out of the tube, the current chip begins to slide, so we will just calculate how long it takes the chip to fall half its diameter and see if we can move our servo back in enough time to catch the next chip. Just a reminder that it takes our servo 30 ms to move back. When the lever moves out, the only force acting on the chip is gravity and we will be using an equation to calculate the amount of time it takes an object moving at v_i to fall y distance in Equations (18)-(20) [15].

$$t = (-v_i \pm \sqrt{(v_i^2 + 2gy)}) / g \tag{18}$$

$$t = \sqrt{(2gy)} / g \tag{19}$$

$$t = \sqrt{(2 * 9.8\ m/s^2 * 0.0195\ m)} / 9.8\ m/s^2 = 0.0631\ s \tag{20}$$

Equation (20) shows that it will take 63.1 ms for the chip to fall the distance of the radius of the chip. This means that the servo will have more than enough time to go back in before the next chip falls through so we can just decrease the speed of this servo and it will be sufficient.

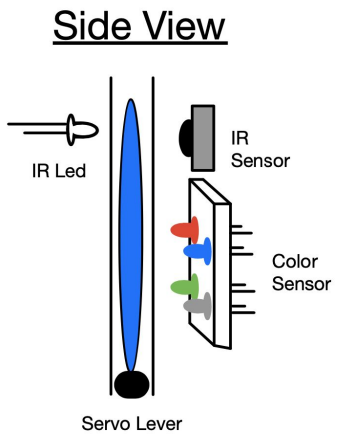


Figure 12. Side view of IR LED, IR sensor, color sensor, and servo lever.

Figure 12 shows a side view of how our sensors will be oriented when our servo lever is in place holding the chip. As the chip obstructs the IR sensor from seeing IR LED, the IR sensor will send a signal to the microcontroller to start the color sensor. The color sensor located underneath the IR sensor will scan the center of the poker chip which is most likely to have differentiating poker chip color. In order to maximize the amount of time the color sensor will have to decipher the color, we will move the servo lever to keep up with the fastest turn over of chips from Equation (8) which is 67.14 ms. Because the color sensor needs 42 ms to process a chip's color as shown in Equation (9), the incoming chips will not clog up the tube.

With this implementation and following these measurements and calculations, our rotating disk will be able to quickly move chips and our servo will be able to stop the chips just long enough for our color sensor to properly get the color intensity values without clogging up the tube.

2.8 Bill of Materials (BOM)

Table 1 shows the expected cost of materials for the project. In total, the project uses 41 components and costs $133.06. All ordered parts come from the following suppliers: Digikey, Sparkfun, Amazon, and PCBWay.

Table 1. Bill of materials for the entire project.

No.	Part	Supplier	Part #	Qty.	Price Per Unit	Total Price
1	5 V Switching Regulator	Digikey	296-43596-1-ND	1	$4.20	$4.20
2	Schottky Diode	Digikey	1727-5841-1-ND	2	$0.42	$0.84
3	47 µH Inductor	Digikey	AIAP-03-470K-ND	2	$1.33	$2.66

Table 1. Bill of materials (continued).

4	100 µF Capacitor	Digikey	PCE3750CT-ND	1	$0.12	$0.12
5	4.7 µF Capacitor	Digikey	PCE4304CT-ND	1	$0.42	$0.42
6	10 nF Capacitor	Digikey	490-13295-1-ND	2	$0.10	$0.20
7	3.3 V Switching Regulator	Digikey	LM2672N-3.3/NOPB-ND	1	$4.20	$4.20
8	330 µF Capacitor	Digikey	PCE3888CT-ND	1	$0.59	$0.59
9	150 µF Capacitor	Digikey	493-2204-1-ND	1	$0.51	$0.51
10	3.3 µF Capacitor	Digikey	399-5502-1-ND	1	$0.25	$0.25
11	Rechargeable 7.4 V Battery	Digikey	1568-1877-ND	1	$15.95	$15.95
12	Power Switch	Digikey	2299-TE6-1A-DC-1-PB-ND	1	$4.16	$4.16
13	Continuous Rotation Servo Motor	Digikey	1528-1496-ND	1	$1.15	$1.15
14	Servo Trigger	Sparkfun	WIG-13872	1	$11.95	$11.95
15	Pushbutton	Digikey	401-1992-ND	1	$12.95	$12.95
16	IR Sensor	Digikey	TSSP4056-ND	1	$0.92	$0.92
17	IR LED	Digikey	511-1363-ND	1	$1.45	$1.45
18	22 Ω Resistor	Digikey	22WCT-ND	1	$0.68	$0.68
19	Color Sensor	Digikey	1738-1035-ND	1	$7.98	$7.98
20	Positional Rotation Servo Motor	Digikey	1528-1075-ND	1	$12.00	$12.00
21	ATMEGA328P-PU Microcontroller	Digikey	ATMEGA328P-PU-ND	1	$2.08	$2.08
22	28 Pin DIP Socket	Digikey	ED3050-5-ND	1	$0.33	$0.33
23	22 pF Capacitor	Digikey	399-1926-ND	2	$0.88	$0.88
24	16 MHz Crystal	Digikey	300-6034-ND	1	$0.54	$0.54

Table 1. Bill of materials (continued).

25	ESP8266 Wi-Fi Module	Amazon	B00O34AGSU	1	$6.55	$6.55
26	2.2 kΩ Resistor	Digikey	CF14JT2K20CT-ND	1	$0.10	$0.10
27	1 kΩ Resistor	Digikey	CF14JT1K00CT-ND	1	$0.10	$0.10
28	LCD Display	Adafruit	181	1	$9.95	$9.95
29	LCD Backpack	Adafruit	292	1	$9.95	$9.95
30	Pushbuttons	Digikey	401-1992-ND	3	$1.10	$3.30
31	10 kΩ Resistors	Digikey	10KQBK-ND	3	$0.10	$0.30
32	PCB	PCBWay	N/A	1	$5.00	$5.00
33	Casing	N/A	N/A	1	$15.00	$15.00
					TOTAL	**$133.06**

2.9 Software-Based Simulation

Figure 13 shows what our dashboard will look like on a player's mobile device as well as the HTML code. Figure 14 then shows pseudocode for controlling the flow of the game and interacting with the preview.

Figure 13. Example HTML code and preview.

```
PlayersList = [] # Holds Total Player's Name, Total Money Amount, Current Bet Amount
PlayersPlayingThisHand = [] # Holds Player's Name, Total Money Amount, Current Bet Amount in each round
# Set Game rules and Logisticss
StartingPlayerTurn = -1
isFinished = false
SmallBlind = SetSmallBlind ()
BigBlind = SetBigBlind ()
isPlayingThisHand = false

# Add players to the game
while (WaitingForPlayersToJoin):
    NewPlayer = AddPlayer (PlayerName, StartingAmount, CurrentBet = 0)
    PlayersList.append(NewPlayer)
UpdateScoreBoard ()

# Main loop that will control the rest of game flow
while (not isFinished):
    # Add all the players playing this hand to PlayersPlayingThisHand List
    for p in range (len(PlayersList)):
        isPlayingThisHand = AskPlayerIfPlayingThisHand (p)
        if (isPlayingThisHand):
            PlayersPlayingThisHand.append (PlayersList[p])

    UpdateScoreBoard ()
    # More logistics
    isSmallBlindSet = false
    isBigBlindSet = false
    # Increment who starts each hand, similar to assigning dealer
    StartingPlayerTurn = (StartingPlayerTurn + 1) % len(PlayersPlayingThisHand)
    CurrentPlayerTurn = StartingPlayerTurn

    for Round in range (1,4): # There are three rounds of betting
        while (isNotDoneBetting):
            # In round 1, the first two players must make small blind and big blind respectively
            if (Round == 1):
                if (not isSmallBlindSet):
                    isSmallBlindSet = true
                    PlayersPlayingThisHand[CurrentPlayerTurn].CurrentBet += SmallBlind
                    PlayersPlayingThisHand[CurrentPlayerTurn].AmountLeft -= SmallBlind
                    (CurrentPlayerTurn + 1) % len(PlayersList)
                    UpdateScoreBoard ()
                    continue

                elif (not isBigBlindSet):
                    isBigBlindSet = true
                    PlayersPlayingThisHand[CurrentPlayerTurn].CurrentBet += BigBlind
                    PlayersPlayingThisHand[CurrentPlayerTurn].AmountLeft -= BigBlind
                    (CurrentPlayerTurn + 1) % len(PlayersList)
                    UpdateScoreBoard ()
                    continue

            # Get the Current player's action
            CurrentAction = PlayersList[CurrentPlayerTurn].Action()
            switch (CurrentAction) {
                # Remove them from PlayersPlayingThisHand List
                case "Fold":    PlayersPlayingThisHand.remove(CurrentPlayerTurn)
                                break;
                # No updates for this player needed
                case "Check":   break;

                # Have them bet the CurrentBet value
                case "Call":    PlayersPlayingThisHand[CurrentPlayerTurn].CurrentBet += GetCurrentBet ()
                                PlayersPlayingThisHand[CurrentPlayerTurn].AmountLeft -= GetCurrentBet ()
                                break;

                # Have them bet the RaiseValue and Update CurrentBet value
                case "Raise":   PlayersPlayingThisHand[CurrentPlayerTurn].CurrentBet += GetRaiseValue ()
                                PlayersPlayingThisHand[CurrentPlayerTurn].AmountLeft -= GetRaiseValue ()
                                SetCurrentBet(GetRaiseValue)
                                break;
            UpdateScoreBoard ()
            # Go to next players turn
            (CurrentPlayerTurn + 1) % len(PlayersList)
            # See if players are done betting
            isNotDoneBetting = SeeIfBettingIsDone ()

    UpdatePlayersList ()
    UpdateScoreBoard
    # See if the players want to play another round
    isFinished = CheckIfFinished ()
```

Figure 14. Pseudocode for controlling the game flow.

3 Second Project Conclusions

3.1 Implementation Summary

Section 1.3 lists out the three most crucial objectives—that is, the high-level requirements—which the overall design must satisfy in order for the machine to work successfully. The entirety of Chapter 2 explains Team 4's design approach and provides justification for why the design meets the necessary objectives. More specifically, Nathaniel and Zach use Sections 2.1-2.5 to specify which components play a crucial part in satisfying the requirements, including the power module, motor driving circuit, poker chip detector, and microcontroller unit. Additionally, TJ gives a mathematical analysis in Section 2.7, showing the feasibility of the machine's design when playing with poker chips. In it, he provides calculations which clarify certain design decisions (such as machine size and sensor placement) and proves that poker chips placed in the spinning drum can and will move through the machine at a rate fast enough to make the machine effective but also not too quickly where it is unable to keep up. And finally, TJ and Nathaniel show a way to structure HTML code for the dashboard (the page displayed in the user's browser) and execution code (in this case, pseudocode) for the machine in Section 2.9. Of course, there are always concerns associated with whether or not a theoretical design will hold up in the real world (Section 3.2 addresses some of these potential concerns), but given the planning and analysis performed to prepare for this design, the expectation is that few problems will arise during the building phase of this project.

3.2 Unknowns, Uncertainties, Testing Needed

Unfortunately, due to unforeseen circumstances, it is not currently possible to utilize the resources on campus to build this project. This largely impacts the project in a couple ways. First, in general, one needs to put forth lots of effort and a significant amount of time to create the physical structure of the console. 3D printing is, perhaps, one of the most effective ways to build parts with great precision, and without access to the fabrication labs, it is much more difficult to build and piece together the mechanical components which make up the console. Seeing as this project relies heavily on proper mechanical design, this is certainly not ideal. Second, the electrical design for this project incorporates the use of a PCB. Without the ability to use soldering equipment in the senior design lab, one must, instead, build the system using a breadboard. Since not all components in the bill of materials are through-hole components, one must, first, find through-hole replacement components for each surface mount part before any circuit construction can begin.These two problems, alone, reduce the overall reliability of the system and add on to the expected amount of time it takes to build the console.

In terms of testing, there are many elements of the design which require experimentation to ensure they work properly. However, the areas which need the most attention relate to the topics discussed in Chapter 2—that is, areas related to high-level requirements. For example, the color sensor must consistently identify the colors of different poker chips. Sometimes poker chips have markings along the outer radius of the chip, and while it is unlikely these markings cause any issues, one must still verify they do not interfere with the color sensor's ability to determine the primary color of the chip. Also, given

that the success of the poker chip detector relies on the system's ability to limit the channel to one chip at a time, one needs to perform multiple trials with the console, varying the amount of chips inserted into the spinning drum each time. And finally, because the third high-level requirement relates to communication between the console and external devices, one must try connecting to the console with a variety of devices and browsers to ensure the Wi-Fi module is compatible with whatever device and/or browser each player uses. Of course, there are many other tests which one can and should perform with the console, but these are the ones which are most important since they directly relate to the high-level requirements.

3.3 Ethics and Safety

There are a few safety and ethical concerns that reside with our project. One of the main safety concerns comes from the batteries that will be used to power the chip reading system and LCD monitor within the main unit. These batteries can explode if they become overcharged or reach an extreme heat temperature [8]. Additionally, these batteries should not be charged in extreme cold temperatures either as they can deteriorate and leak chemical acid [8]. To ensure safety, we will adequately test and monitor the battery cell temperature to ensure its overall quality and performance.

With our project design containing electronics like the IR sensor and the color sensor, the safety issue of exposed wires arises. Exposed wires can cause detrimental damage to both the other electronics surrounding it and the users with burns and shocks being common injuries. To safely and effectively avoid the safety concern, we plan to insulate the wires through the use of electrical or thermal insulating tape. Additionally, we will also verify our electronic components are being supplied with the correct amount of voltage and current going through each of them by testing each individual part and seeing if it meets the necessary requirements/limits. To ensure our wires are safely insulated, we will test them and ensure they do not conduct any electricity as well as not heat up to dangerous temperatures.

Another safety concern stems from the motor driving circuit. This mechanism will utilize servo motors that will exert force on a spinning disk to push the received chips to the outer edge of it before going down a channel chute. The concern that arises is possible injury from these motors when inserting a chip into the mechanism. To combat this safety concern, we plan on abstracting away the motor driving circuit from the user. We plan to have a funnel that opens up wide enough for a player to insert multiple chips at once. This funnel will be raised high enough that a person will not be able to stick their finger through the funnel and reach the rotating disk.

An ethical concern our project raises involves the containment of private user data. We plan on utilizing an HTML website to maintain, distribute, and update game data for players. Some of this data includes the player's amount of money remaining and the current amount of the pot for that round. This issue raises the concern of possible data leaks or piracy that can take place through malicious software attacks focused on web application data. These issues go against the IEEE Code of Ethics #9 - "to avoid injuring others, their property, reputation, or employment by false or malicious action" [9]. To mitigate this issue, we will implement a web application firewall that will check all incoming traffic and prevent any incoming, malicious attacks. Additionally, we plan on encrypting the game data as well for an extra layer of protection against attackers. Finally, we will provide a notification to users to remind them to

check to make sure their external device has no malicious software embedded into it that can potentially compromise the website.

Since the game of poker involves the aspect of money, both the mobile device and LCD display should reflect the correct amount of money that each player currently has as well as the correct amount of money in the pot. Our project raises the concern these values can possibly be skewed. This concern violates the IEEE Code of Ethics #3 - “to be honest and realistic in stating claims or estimates based on available data” [9]. To avoid violation, we will accurately store the correct user inputted data as well as perform correct arithmetic on these values. Lastly, we will effectively update these now-changed values and correctly display them on the proper interfaces, so users can properly see them.

Finally, to avoid violation of IEEE Code of Ethics #1 - “hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment” [9]; we accept responsibility for our design. Our project aims to ensure the safety of its users and while we hope to test all possible scenarios there are a billion more that may occur. Specifically, we want to ensure users are completely safe from the mechanics within the main unit like the chip reading system. Therefore, we accept responsibility for faults in our design and ensure we take appropriate action to ensure better user safety when these faults occur by updating our design.

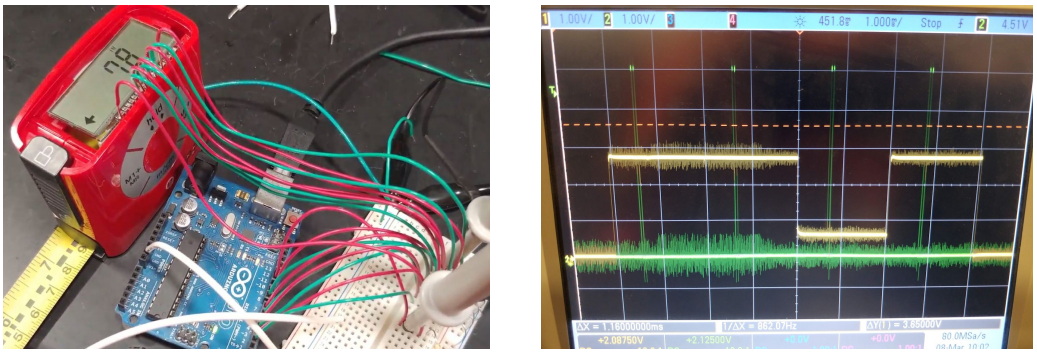
3.4 Project Improvements

Given the time constraints of ECE 445 as well as some of the design requirements for the course, this project, no doubt, is basic in nature and has room for improvement. With this being said, here are a few suggested revisions for those who may try implementing this project in the future. The first two suggestions relate to minimizing cost and reducing the number of components needed inside the console. As it currently stands, the microcontroller used in this design needs a separate Wi-Fi module in order to communicate with external devices. With this being said, one can buy an ESP32 which acts both as a SoC and a Wi-Fi module at a lower cost than the combined price of an ATmega328P and an ESP3288. Also, not only is it cheaper, but it is faster, has more memory, and even has multiple cores. This leads into the next suggestion: get rid of the servo trigger. The reason the design incorporates the servo trigger in the first place is because it reduces the load on the microcontroller. Because the ESP32 has two cores (unlike the ATmega32P which only has one), it can use its extra core to perform the same function as the servo trigger. This, again, takes another component out of the design, reducing costs as well as the number of components. The final recommendation does not require any additional hardware or extra purchases. The original intent of this project was to build a machine which helps people play the game of poker, but there are plenty of other games one can play with a deck of cards. So why not design the console to work with those games, too! The current design already manages scores, so with just a little more work on the software side of things, one can enhance the functionality of the machine to incorporate a wide number of other games as well. By implementing these features and modifications, the price of the console will, no doubt, go down, and ultimately, the overall satisfaction derived from using the product will go up!

3.5 Progress Made On First Project

Before beginning the poker chip project, Team 4 started the semester with a different project titled, “Haullelujah! A Solution to Packing a U-Haul!” The concept of the project was to develop an app to aid users who are in the process of moving by giving recommendations on where to place boxes within a U-Haul to maximize space and reduce damage to items during transit. Along with the app, Team 4 planned to create a Bluetooth-enabled digital tape measure which pairs with the device running the application to speed up the process of measuring dimensions of boxes. Digital tape measures already exist, but there are none that work with an app like what Team 4 was trying to make. So, the original plan was to “hack” into a digital tape measure which is already on the market, use a microcontroller to decode signals coming from the tape measure’s display, and then send this information to the mobile device so one does not have to manually input dimensions within the app (which takes time and requires lots of repetition).

Originally when this idea was proposed, there was speculation as to whether or not this method of hacking into a digital tape measure’s display would actually work. The course staff recommended getting an early start, so Team 4 began working in the senior design lab the week before spring break to experiment with the tape measure they ordered the week before. They took it apart, removed a protective cover on the screen, reassembled the tape measure, and soldered wires onto its terminals. Next, they connected the wires coming from the display to an oscilloscope. Trying different combinations of wire pairs, eventually, the team figured out which terminals controlled which numbers on the display. Then, by stretching the tape measure to different lengths, they figured out how the serial communication to each pin worked. And finally from there, they began using an Arduino to fully decode the signals coming from the display. Figure 15 shows how the tape measure was physically set up for testing as well as an example of the readings taken by the oscilloscope from the Arduino, capturing the signals from the tape measure’s display (where the green line’s spikes indicate Arduino digitalRead commands and the yellow line is the original signal from the tape measure). Although the team did not get the chance to complete the project, they enjoyed the overall process of coming up with the idea and are confident that, had they been able to finish the semester on campus, they could have built a fully-functioning system.



(a) (b)

Figure 15. Tape measure (a) testing setup and (b) oscilloscope readings from display.

References

[1] Gabi, “6 Most Popular Card Games in the World,” Bridge Is Cool, 25-Apr-2019. [Online]. Available: https://bridgeiscool.com/card-games-2/. [Accessed: 30-Mar-2020].

[2] U. Chavan, A. Giridharan, and V. Pitta, “Electronic Betting System For Poker.”

[3] B.Pempus, Las Vegas Poker Rooms Report Strong February Revenue. [Online]. Available: https://www.cardplayer/com/poker-news/22659-las-vegas-poker-rooms-report-strong-february-revenue. [Accessed: 30-Mar-2020].

[4] B.Ray, “RFID Vs. Wifi: Comparing The Technology & Costs For Asset Location,” AirFinder, 07-Feb-2017. [Online]. Available: https://www.airfinder.com/blog/rtls-technologies/rfid-vs-wifi-comparing-technology-and-cost.[Accessed: 30-Mar-2020].

[5] “Bold Poker,” BoldPoker - Deal with it. [Online]. Available: https://www.boldpoker.net/. [Accessed: 30-Mar-2020]

[6] Casino poker game table that implements play of a casino table poker game, Feraidoon Bourbour, Troy Nelson. (2008, Mar. 8). *Patent US7341510B2*. Accessed on: Apr. 3, 2020. [Online]. Available: https://patents.google.com/patent/US7341510B2/en?inventor=Feraidoon+Bourbour

[7] Electronic video poker method and system having multiple poker hands, by Mark Angel. (2004, Feb. 24). *Patent US6695695B2*. Accessed on: Apr. 3, 2020. [Online]. Available: https://patents.google.com/patent/US6695695B2/en

[8] batteryuniversity.com,“Lithium-ion Safety Concerns,” *Lithium-ion Safety Concerns–Battery University*. [Online]. Available: https://batteryuniversity.com/learn/archive/lithium_ion_safety_concerns. [Accessed: 07-Feb-2020].

[9] Ieee.org, “IEEE IEEE Code of Ethics”, 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 03-Feb-2020].

[10] *Antique coin counter by Jeffery A. Krueger*. 2010. [video] Directed by J. Krueger.

[11] “Poker Chip Sizes,” *Sidepot Clay Poker Chips*. [Online]. Available: https://www.buypokerchips.com/Articles.asp?ID=298. [Accessed: 17-Apr-2020].

[12] "TCS3200 Color Sensor (SKU:SEN0101)," *DFRobot*. [Online]. Available: https://media.digikey.com/pdf/Data Sheets/DFRobot PDFs/SEN0101_Web.pdf. [Accessed: 17-Apr-2020].

[13] "Topic: How long should digitalWrite() take?," *Arduino Forum*. [Online]. Available: https://forum.arduino.cc/index.php?topic=41046.0. [Accessed: 17-Apr-2020].

[14] "Sweep," *Arduino*, 18-Aug-2015. [Online]. Available: https://www.arduino.cc/en/Tutorial/Sweep. [Accessed: 17-Apr-2020].

[15] R. Kurtus, "Gravity Time Equations for Falling Objects," *Gravity Time Equations for Falling Objects by Ron Kurtus - Physics Lessons: School for Champions*, 05-Jan-2011. [Online]. Available: https://www.school-for-champions.com/science/gravity_equations_falling_time.htm#.XplbbFNKjOQ. [Accessed: 17-Apr-2020].

[16] "Continuous Rotation Servo - FeeTech FS5103R," *Adafruit*. [Online]. Available: https://media.digikey.com/pdf/Data Sheets/Adafruit PDFs/154_Web.pdf. [Accessed: 16-Apr-2020].