ChipCaddy: A Home Poker Game Solution

ECE 445 Design Document

Project #16

Marvin Camras, Anish Rajesh, Justin Wang

Professor: Olga Mironenko

TA: Nikhil Arora

Introduction

Problem

According to a market research study published by Zion Market Research, the demand analysis of Global Trading Card Game Market size & share revenue was valued at $6.39 Bn in 2022 and is estimated to grow about $11.57 Bn by 2030 [3]. Although gambling has its pitfalls, it has become one of the world's most predominant pastimes and as a result created a market of equal size. Whether at a casino or in less organized settings, games like poker, roulette, and blackjack bring individuals together all over the world in social settings allowing them to build rapport, have fun, and have a chance at making a profit. As the market for card games increases, so does the need for accurate, secure, and efficient home game systems. Current home games are set up with a simple set of chips, cards, and players, resulting in large amounts of time wasted counting, sorting, and dealing chips. Casinos are well equipped with the endowment to purchase top-end counting mechanisms such as RFID poker chips or table-embedded chip counting mechanisms, but these machines cost thousands of dollars on average and are not suited for the casual home game.

Games such as Omaha are pot-limited, meaning the max bet players can make is the amount of chips currently in the pot. With the current home game system, players must hand count the amount of chips currently in the pot, as well as manually sort and dispense chips after each and every hand. This results in not only a large amount of time wasted, but also makes it easy for players to steal chips and miscount the current value of the pot.

In addition to this, calling players all-in values requires manually counting each stack of chips by hand, which can lead to incorrect values and a lot of wasted time. Online games have an automatic display of each player's stack, resulting in almost 3 times faster gameplay according to Upswing Poker [3].

Solution

In an effort to promote ease of play, and maximum efficiency in the number of hands played, we have come up with a solution featuring a combination of sensors, motors, and internal logic to sort poker chips and display the current value of the pot for all players. After a hand, the chips will be inserted as a stack into a rotating tunnel. The tunnel will have a color sensor at its base, as well as a slit for the chip to be shot out of using a motor. The color sensor will relay the color of the chip to the microcontroller, which will handle the logic and display the current pot value on an LCD display based on the respective colors. As soon as the color of the chip is detected, the tunnel will rotate to a designed container for that color. As the chip is ejected, the microcontroller will keep track of the chip color ejected, and the total number of chips ejected to make sure none are lost. When the winner of a pot is determined, the pot winner will retrieve their chips from the containers and hit the reset button so the pot count on the LCD display resets to 0 for the next hand. In the rare case of a split pot, the user will be able to press a button that displays the respective color denominations for each split of the pot. The microcontroller will be responsible for the logic portion and will sit on a PCB that is also used to power our contraption. The device will also use a series of buttons to take in user information regarding big blind, small blind, and buy in values.

Our solution will greatly increase the efficiency and enjoyment of a regular home poker game. Rather than wasting valuable time counting pots, distributing plastic chips, and arranging chips in neat color coordinated towers, players can focus on having fun and playing.

Visual Aid

We will be utilizing a CNC rotating base from the machine shop for our project. This metal base features a 360 degree servo motor bolted in, allowing us to have a robust design without having to 3D print the device's foundation.

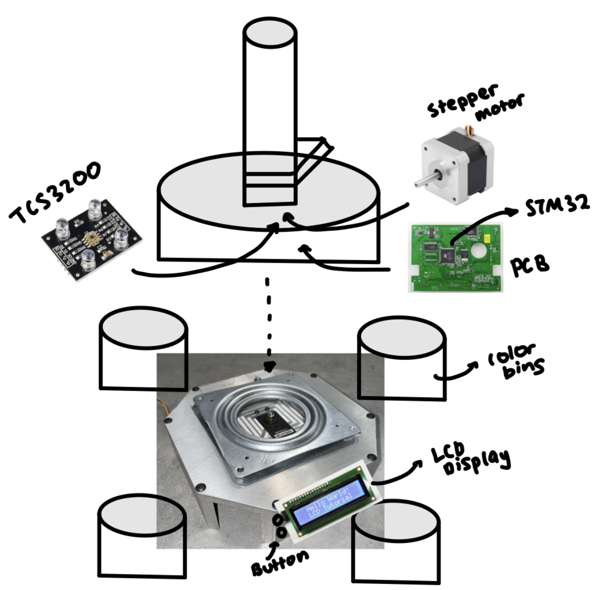


Fig. X - Simple visual aid of our device

When the unsorted chips are placed into the entry funnel the color sensor uses infrared technology to sense the color of the bottom chip. The STM32 microcontroller will then rotate the device to point towards the respective color bin. Once facing the bin, the linear actuator will push the bottom chip through a slit in the funnel (big enough for only one chip at a time) into its respective bin. The LCD display will then append the count of the pot based on the color of the chip. This process repeats until the color sensor detects that there are no more chips left in the funnel. In the case of a split pot, the user will be able to press the button next to the display.

High Level Requirements

In order for our project to be deemed a success, it must meet the following requirements:

1. **The device should append the count within 5 seconds of the chip being read by the color sensor.**
2. Upon ejecting all chips from our contraption, the winner of the pot will be able to **reset the pot count to 0.**
3. In the case of split or chop pots the user will be able to manually **choose the number of ways the pot will be split, and the respective color denominations (White- $0.25, Red- $0.50, Green- $1, Black- $2) for these divisions will be shown on the LCD display.**
4. The device will keep a tally of the number of chips counted, and the number of chips counted for each color. **It will then ensure the sum of the number of each colored chip matches the total number of chips and display a warning on the LCD if not.**

Design

Block Diagram

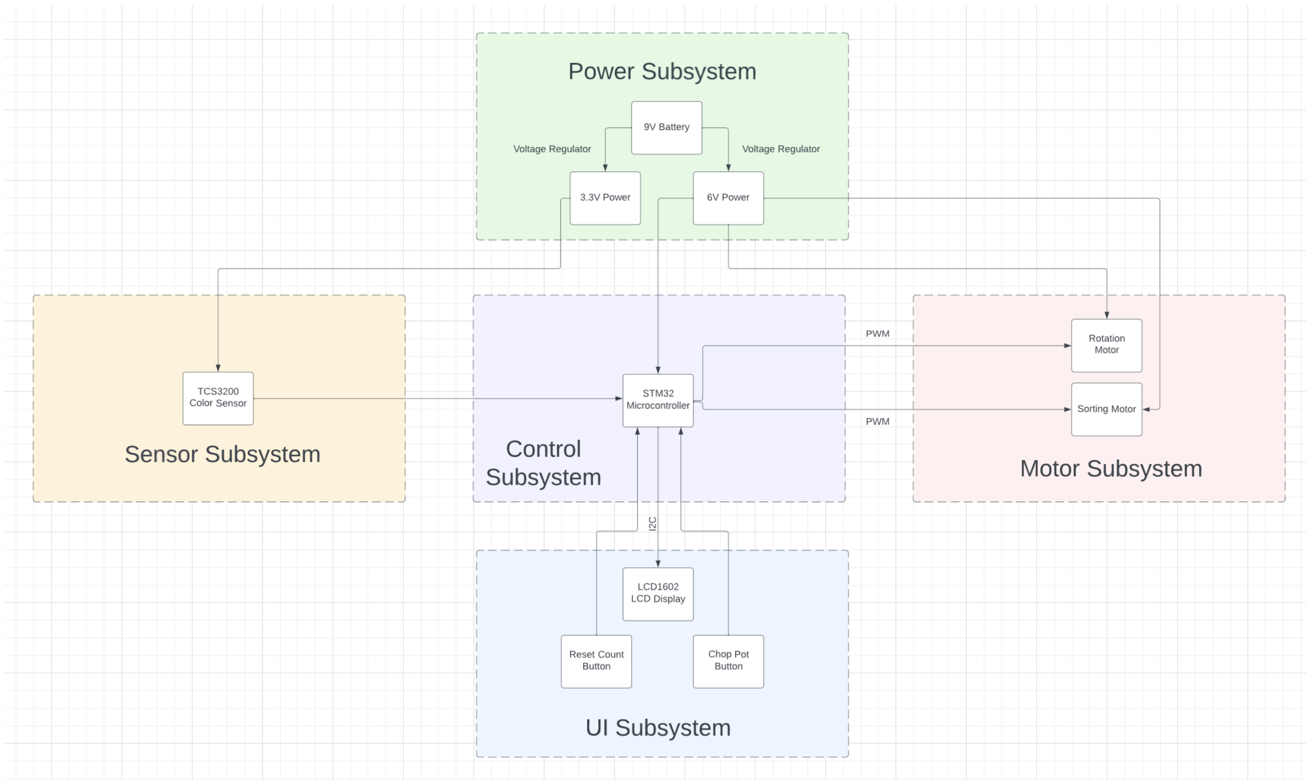


Fig. X - Block diagram of our device with all subsystems

Our high level block diagram features all the subsystems and components that will be featured in our design. The design is powered by a standard 9V battery. This battery will be stepped-down to 3.3V and 6V power with voltage regulation on our PCB. The 6V power will then be used to power the STM32 microcontroller and our servo motors. The STM32 will serve as the brains behind our design, handling all the computation and telling the motors how much to rotate based on what color the TCS3200 color sensor sends. Based on what the sensor subsystem sends to the control subsystem, the motors will place the correct color chip in its respective bin. The control unit will then send the correct value of the pot to our LCD display through I2C serial communication. Once the hand is over, the user will be able to interact with the UI subsystem through two buttons, one for chopping the pot - which will send the correct value of the split onto the LCD display - and one for resetting the value of the pot back to 0.

Physical Design

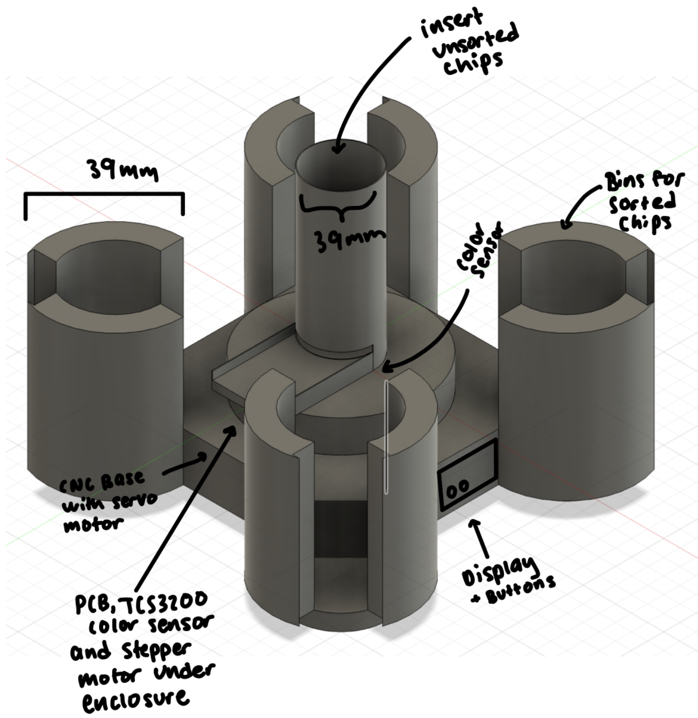


Fig. X - Fusion 360 CAD model of our device

The device will be CNC cut with help from the machine shop. The rotating base will be retrofitted from a previous semesters project. The chip bins will be ~ 39mm in diameter, which is roughly the standard size for commercial poker chips. A PCB featuring the STM32 microcontroller will be placed directly under the enclosure, along with a linear actuator and the IR color sensor. Due to the IR technology utilized by the sensor, it does not matter that there is a low level of light in the funnel. The chips will be placed into the funnel, and the color sensor will then send the color signal to the MCU. The device will then rotate so that the opening of the funnel is facing the correct color bin, and the linear actuator will then shoot the chip into its correct bin.

Subsystems

Power Subsystem:

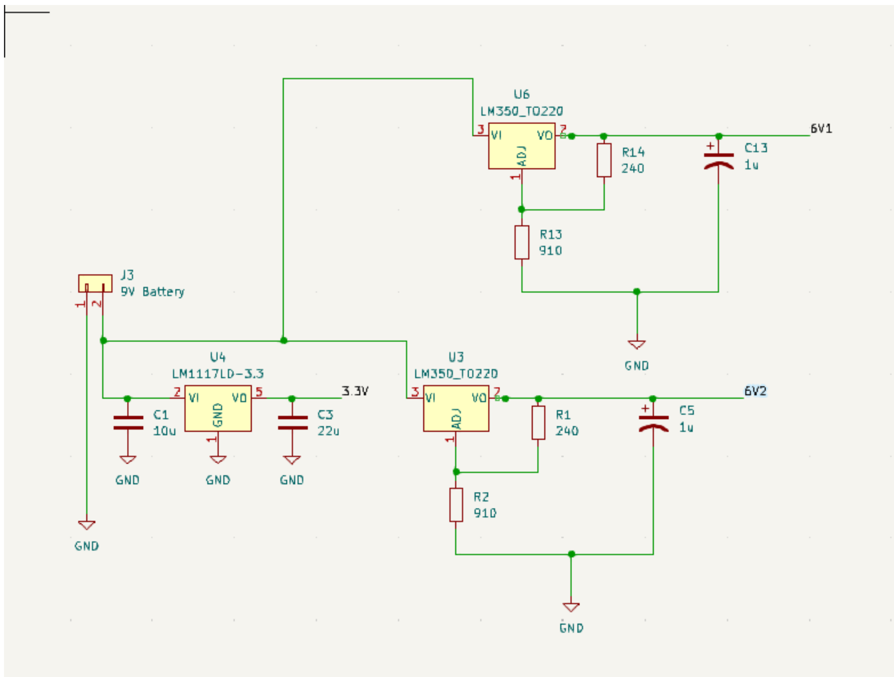


Fig. X - KiCAD schematic of our power subsystem

Our device will be initially powered with a replaceable 9V battery. This design choice was made to avoid any need to plug into a socket, as our device is made to be a modular extension of a home poker game. The need to find an outlet and avoid tripping over a large wire is something we wanted to avoid with our design. The electrical components in our design will require 6V (motor subsystem) and 3.3V (UI subsystem, control subsystem, and sensing subsystem) power, so we will need to utilize voltage regulators to step down from our initial 9V input. The 3.3V regulator was a straightforward design choice - the AZ1117 linear voltage regulator features a fixed 3.3V output. We decided to use the LT1117 linear voltage regulator which serves as a direct alternative to the AZ1117. We utilized the design recommended in the device's datasheet as pictured below.

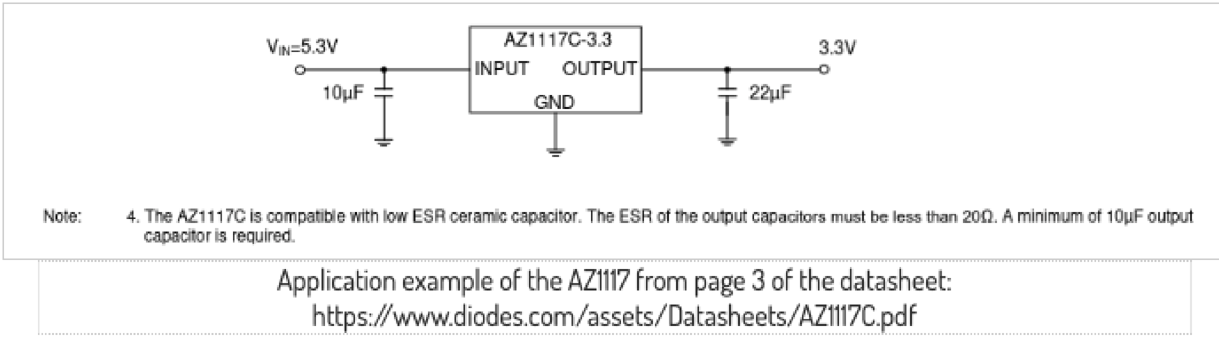
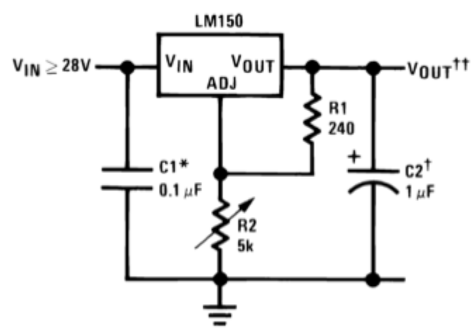


Fig. X - AZ1117C voltage regulator example from ECE 445 Wiki

Next was designing the voltage regulator for our 6V power buses. This was slightly more complicated than that of the 3.3V as there was no fixed regulator available. We utilized the LM350 adjustable regulator due to the extensive documentation available online. To determine the exact value of resistors needed for the design we turned to the datasheet again.



$$\dagger\dagger V_{OUT} = 1.25V\left(1 + \frac{R2}{R1}\right) + I_{ADJ}\,(R2)$$

Fig. X - LM350-ADJ datasheet example schematic

Our design will require a 3V step down from our 9V input. We utilized the equation featured on the datasheet to calculate the value of R2 required for our design.

$$V_{OUT} = 6 = 1.25\ (\ 1 + \frac{R2}{121})$$

Solving for R_2 gives us 910 ohms. This completes the schematic for our 6V power bus. This will be used to connect to our motor subsystem, which will be controlled by our STM32 microcontroller through PWM to be able to rotate the chips to their correct bin and dispense them. We then verified our design through an LTSpice simulation that is pictured below. As we can see, the input is a 9V battery and the output signals are at 3.3V (blue signal) and 6V (red signal). For our motor components we will be using a 360 degree continuous servo motor for rotating the base and a linear actuator to perform the pushing motion on the chip. Both of these parts require a rather high maximum current draw (both over 500mA) so to be on the safe side and to ensure that we supply enough current without overheating the voltage regulators we will need two LM350s to do the job. Although one LM350 can supply 3A of current, since we are stepping down 3V for the motors (9 => 6), the power will be too high for one regulator and it will overheat. To solve this, we will have to generate two separate 6V power buses in order to ensure the operation of our motors without overheating.

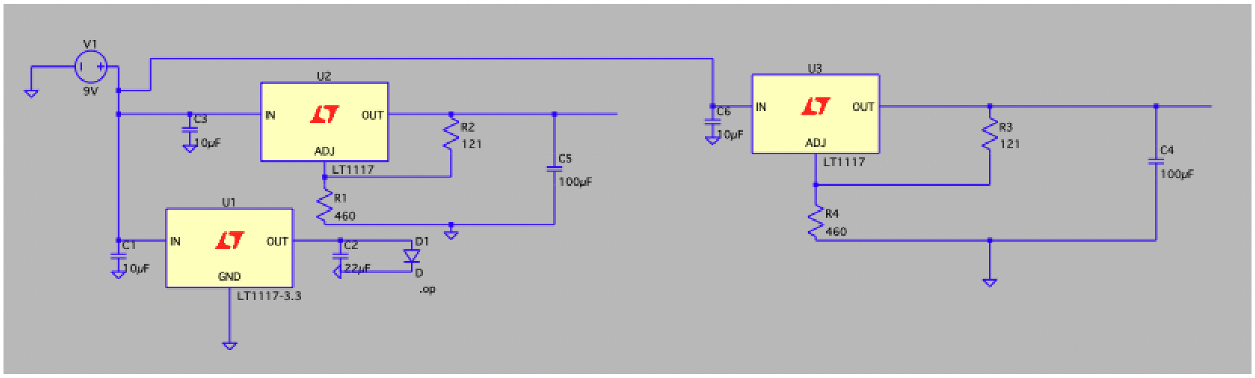


Fig. X - LTSpice operating point simulation of our KiCAD schematic

Requirements	Verification
1. LT1117 Voltage Regulator provides 3.3 +/- 0.5% V Output. Each LM350 Voltage Regulator provides a 6 +/- 0.5% V Output.	1. Use a multimeter and measure the voltage at both of the output nodes of the voltage regulators to verify that they are supplying within 0.5 % of 3.3 and 6 V. 2. Use the multimeter to measure the voltage at inputs of each device to verify that the voltage supply is being supplied correctly as well as to make sure that we have a stable connection between the Regulator and the devices. 3. Tabulate all of the measured values to ensure that each voltage output is accounted for as well as each input to the various devices.
1. LT1117 Voltage Regulator supports at least 75 mA of current. Each LM350 Voltage Regulator supports a minimum of 2.5 A.	1. Use a multimeter at the output nodes to measure the output current at each of the voltage regulators.
1. LT1117 Voltage Regulator stays under its maximum junction temperature of 150° C. Each LM350 Voltage Regulator stays under its maximum junction temperature of 125 ° C.	1. Take an infrared thermometer and measure the surface temperature of the three linear voltage regulators and ensure that the temperatures are under 125 ° C (150 ° C for the LT1117).

<u>Control Subsystem</u>

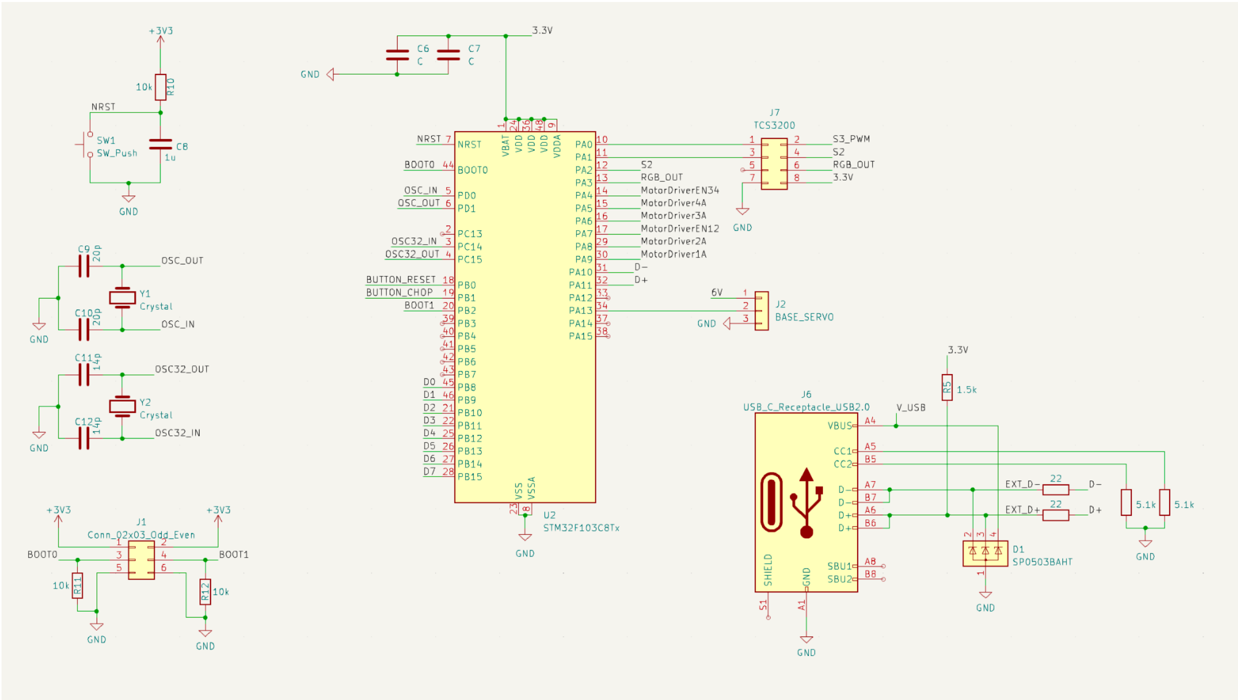


Fig. X - Control subsystem KiCAD schematic

The control unit is the most important part of our design, and it is paramount that we make sure we get the design correct. We decided on the STM32 microcontroller for our design. We originally selected the ESP32 MCU, but pivoted from this design choice due to the fact that we did not need the extensive Bluetooth and Wifi capabilities of this chip. The STM32 has extensive resources available, and features an intuitive IDE through STM32duino that compliments our group members' Arduino experience.

Featured above is the schematic for our microcontroller. The microcontroller will interface with all of our other subsystems, so it is important to understand the relationship between them all.

The control unit directly controls the motors in our design. The servo motor in the base of our device was repurposed from a previous semesters group, and can directly accept a 6V input. This will bypass the need for an H-bridge, and the motor can be directly connected to the pins of the microcontroller, which will then use PWM to tell the motor how much to rotate based on the color input it receives from the sensing subsystem. The dispensing motor will be connected to the microcontroller through a motor driver in order to regulate the amount of voltage it receives.

The sensing subsystem features a TCS3200 infrared color sensor. This is featured through a general connector in our PCB schematic. The TCS3200 will then interface with the microcontroller and tell the MCU what color is being detected.

The UI subsystem features two buttons - one for resetting the count and one for split pots - and an LCD display. The buttons are connected directly to the GPIO pins of the MCU, and when the button is pressed the MCU will perform the necessary calculation and display the result onto the LCD screen.

In the design of our device, which employs a combination of motors and an IR color sensor to efficiently sort chips into their color-specific bins, we have chosen to incorporate a crystal oscillator for critical clock timing. This is featured to the left of our schematic. This decision is driven by the imperative need for precision timing to coordinate the intricate sorting process. The crystal oscillator ensures that the STM32 microcontroller, at the heart of our device, receives highly accurate and stable clock signals, allowing for precise synchronization of motor movements and real-time data acquisition from the IR color sensor. Additionally, the oscillator's exceptional frequency stability, even in the face of environmental variations, guarantees consistent and error-free operation, ultimately enhancing the reliability and efficiency of our chip sorting system.

The other components in our schematic come directly from the STM32 example schematic from the ECE 445 Wiki site. The USB module will be used to program our MCU.

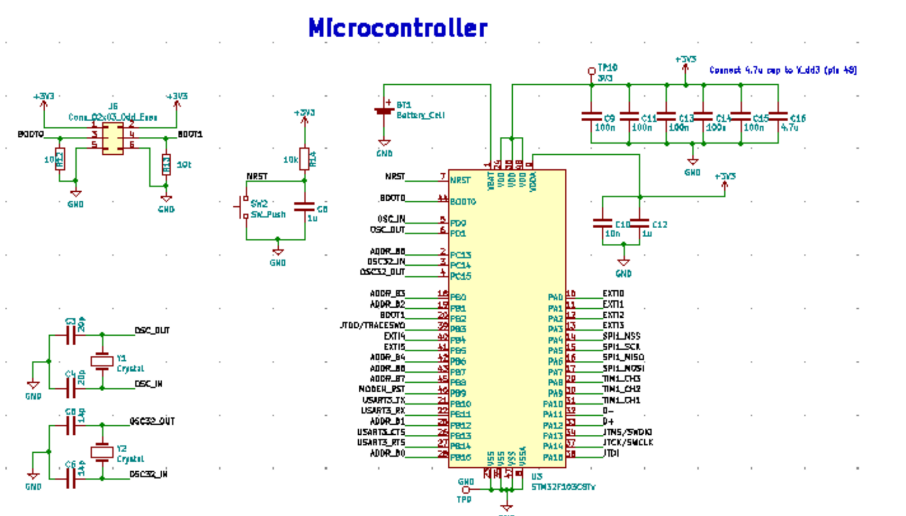


Fig. X - Example schematic from ECE 445 Wiki page

Requirements	Verification
1. Microcontroller is able to analyze data that is received from the TCS3200 Color Sensor. MCU then uses this data to rotate the base motor and push out the chips once we are directed to the correct bin.	1. Once the MCU knows the color on the chip that we are dispensing, we can then confirm the operation of the motors visually by verifying that first, the base motor rotates the pusher in the correct direction and after that, the pusher motor should dispense the correctly identified chip into the bin. The order of this operation within the MCU must be as follows: Analyze the Color => Rotate Base Motor => Shoot the chip out into the bin using the pusher motor.
2. Microcontroller is able to perform the logic required to count the pot, and organize the colors as expected.	1. We can verify accurate calculations and communication with the LCD display by manually counting the value of chips in an inserted stack and seeing if that number is relayed to the display after sorting. We can then test the divisions the same way, do the calculations by hand and see if the LCD display matches that value after hitting the split pot button. 2. Verifying that the colors are assigned to the correct bins can be verified visually.
3. Ensure that we are receiving a 3.3 +/- 0.1 V supply into the MCU.	1. Use a voltmeter at the Vdd input pin to ensure that we are getting a 3.3 V supply into the MCU. Perform the measurement while sorting and dispensing a full chip stack to

	ensure that we have a stable voltage supply throughout the full operation.

Sensing Subsystem

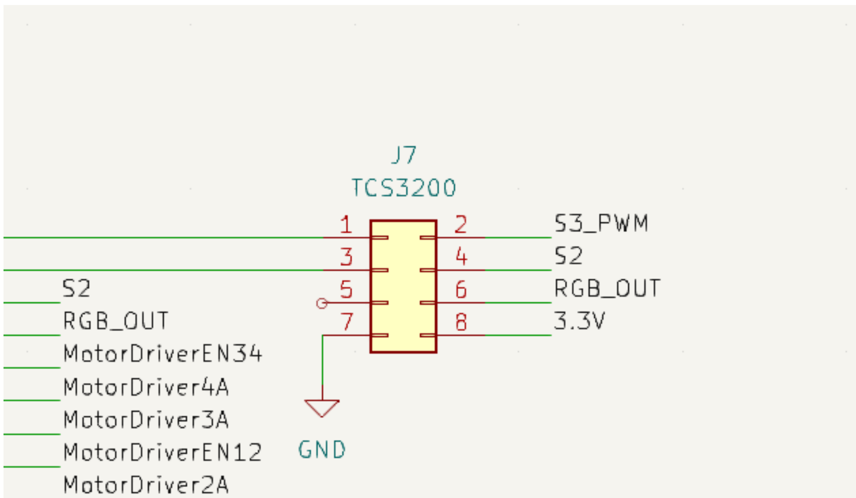


Fig. X - TCS3200 Color Sensor KiCAD schematic

The sensing component of our senior design project will be accomplished through the TCS3200 color sensor.

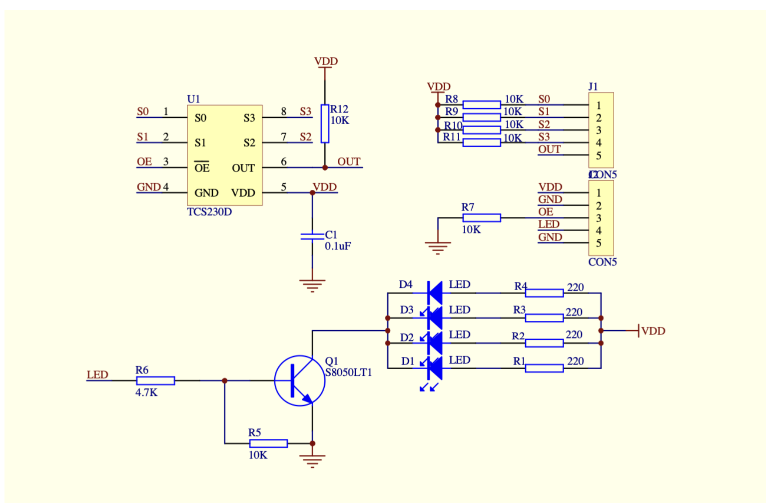


Fig. X - Circuit schematic of the TCS3200 color sensor

The TCS3200 color sensor operates using an array of photodiodes with color filters (red, green, blue, and clear) and an integrated white LED light source. When the LED illuminates an object, the photodiodes detect the intensity of specific color components and convert this into square wave frequencies. These frequencies, proportional to the color intensity, are then counted by our STM32 to determine the object's color. By adjusting the integration time, the sensor's sensitivity can be optimized for various lighting conditions. This will be invaluable for our design as the color sensor will be placed in a dim environment under a stack of chips. The MCU will then accept the color input and tell the motors to rotate and dispense accordingly.

Requirements	Verification
1. The microcontroller receives the right RGB value corresponding to the chip that is inserted, based on information relayed from the TCS3200 sensor. (As long as the read value is closer to the corresponding color than the color of another chip it is acceptable).	1. Insert the maximum amount of chips the contraption supports , of varying colors. 2. Using serial debugging, record the values received by the microcontroller that corresponds to the TCS3200, ensuring that each value is closest to that of the respective color. E.g. if the RGB value for red is 320, the read value is 300, and all other colors are further away from 300 it is acceptable. 3. Record 2 trials detailed by steps 1 and 2 and record the color of the chips, what their expected RGB values were, and what they actually were. *Note: Serial Debugging refers to hooking the STM32 up to a computer using the corresponding TX and RX hardware pins and running an emulation software such as PuTTY or Arduino Serial Monitor.*

1. The TCS3200 sensor receives between 3.3 +/ 0.5% V from the power subsystem.	1. Insert a singular chip into the contraption. 2. Power only the sensor and use a voltmeter to measure the voltage supply to the sensor. 3. Repeat steps 1 and 2 for all four colors, and record data in a table.

<u>Motor Subsystem</u>

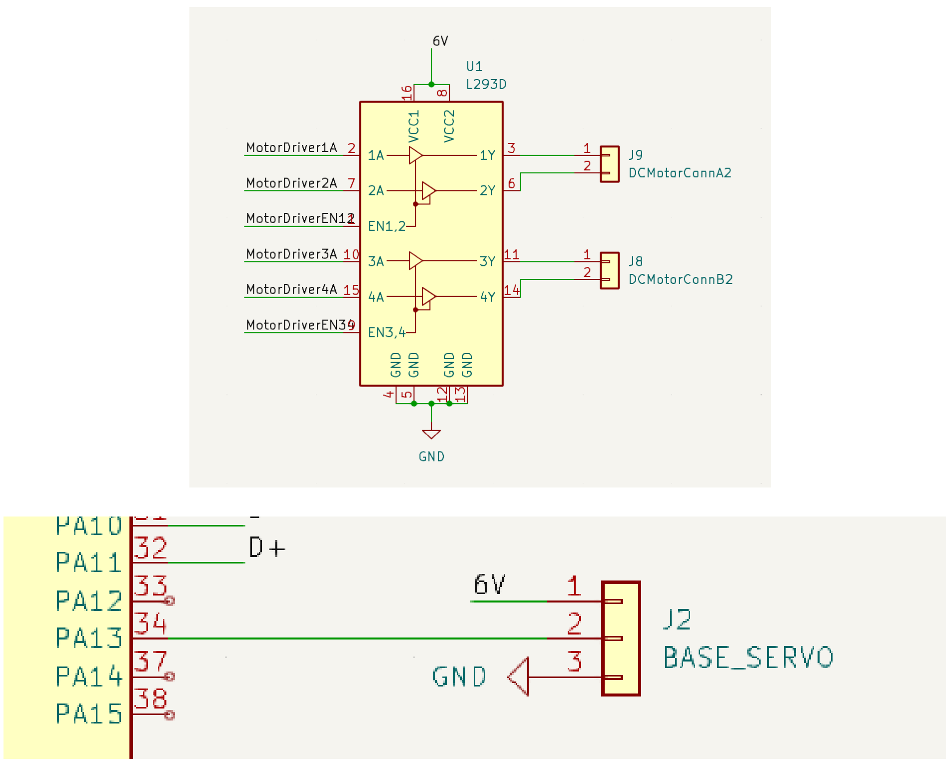


Fig. X - Motor subsystem KiCAD schematic

The motor subsystem is responsible for the movement of our device. The design will feature one motor in the base of the device and one linear actuator at the base of the funnel to dispense chips into their respective bins. The above KiCAD schematic describes our design. The servo motor at the base can accept 6V input, and thus will not require a motor driver to connect directly to the microcontroller. The servo will be connected directly to the GPIO pins of the MCU and will accept information on how much to rotate based on the color that the sensing subsystem detects.

The “pushing” motor will require a motor driver as it cannot directly accept its 3.3V operating voltage directly from the MCU. We selected the L293D motor driver as it has extensive documentation and has been used by previous groups in the past with success. The L293D is a 16-pin Motor Driver IC which can control a set of two DC motors simultaneously in any direction. The L293D is designed to provide bidirectional drive currents of up to 600 mA (per channel) at voltages from 4.5 V to 36 V. The bidirectionality of the driver will allow us to have 360 degree control of our motors and let us optimize sorting time by programming logic for the motor to take the shortest circular path from color to color. We will be using a linear actuator to push the chips out at the end of the funnel. The linear actuator will be connected from the general connector pins that are connected at the end of our motor driver.

Requirements	Verification
1. Both motors receive 6 +/- 0.5% Volts from the power subsystem.	1. Insert three chips into the contraption. Make sure the three chips have at least one adjacent rotation and one 180 degree rotation. 2. Apply a voltmeter to both motor connections and record the values in a table for all the chip ejections.

<u>User Interface Subsystem</u>

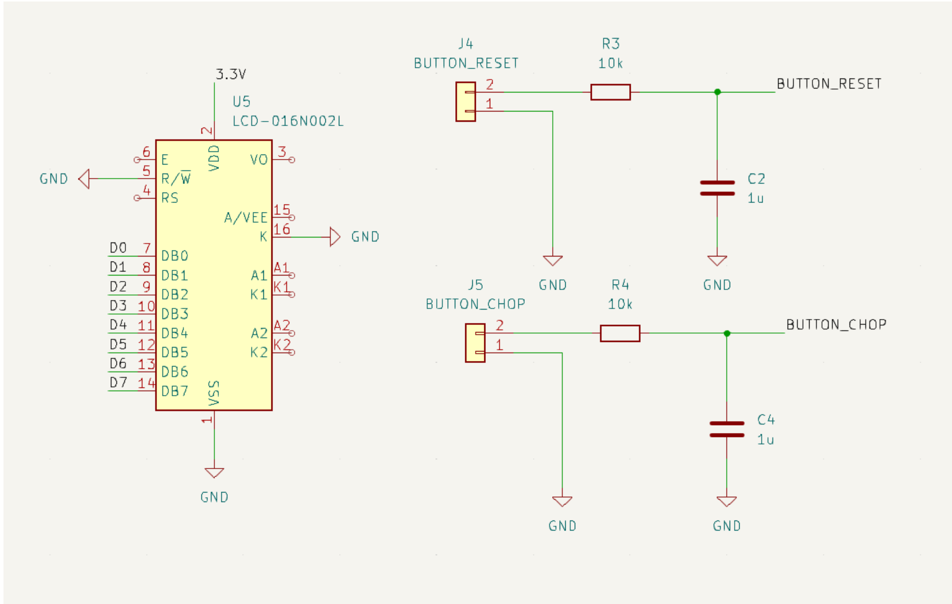


Fig. X - User Interface KiCAD schematic

The user interface subsystem is the user's direct interface with the device. The LCD display is connected to the MCU directly - the pin assignments were based on existing designs from EasyEDA, featured below.

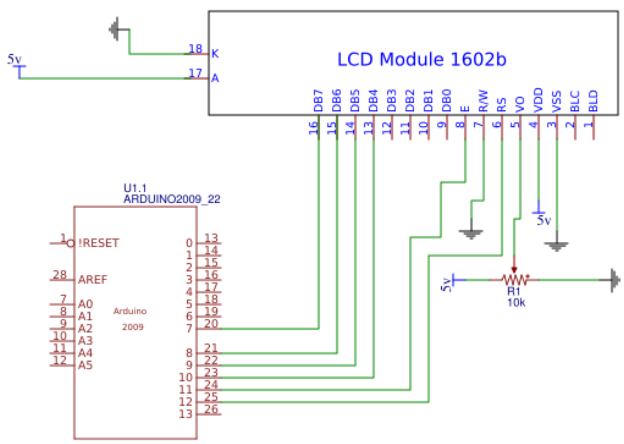


Fig. X - EasyEDA LCD-1602 pin assignments

We selected this LCD display due to its history of use in Arduino and other MCU projects, and due to its ability to display a large amount of text which will be needed in our design when displaying pot values.

The buttons will allow the user to directly input commands into the MCU. Both buttons will directly connect to the GPIO pins of the MCU. When the button is pressed, the MCU will send data to the LCD display to show. We could implement some kind of debouncing circuit to our buttons to prevent any noise, but since our buttons will only be used once per hand we decided against including this in our design.

Requirements	Verification
1. The remote microcontroller must be able to detect the press of the 'reset' button in at most a second of the press or less.	1. Using serial debugging, record the value received by the microcontroller when the 'reset' button has not been pressed. 2. Next, press and hold the 'reset' button. Using serial debugging, record the value received by the microcontroller, and ensure it is different from the value when unpressed. This should occur in at most a second, a stopwatch can be utilized to ensure this. 3. Finally, release the 'reset' button. Record the value received by the microcontroller using serial debugging. Ensure that this value is the same as the original unpressed value. 4. Organize all recorded values in a table. *Note: Refer to the sensing subsystem section for information regarding serial debugging.*
2. The remote microcontroller must be able to detect the status of the 'chop' button in at most a second of the press.	1. Using serial debugging, record the value received by the microcontroller when the 'chop' button has not been pressed.

	2. Next, press and hold the ‘chop’ button. Using serial debugging, record the value received by the microcontroller, and ensure it is the same as the value when unpressed (or previous value for +1 iterations). 3. Next, release the ‘chop’ button. Using serial debugging, record the value received by the microcontroller, and ensure it has changed from the unpressed default value.This should occur in at most a second, a stopwatch can be utilized to ensure this. 4. Continue performing steps 2 and 3 for three more iterations, ensuring that the value received by the microcontroller changes only upon the release of the ‘chop’ button using serial debugging. On the second of these two iterations, the value received by the microcontroller should match that of the default unpressed value of the ‘chop’ button. (Our device accommodates up to a 3 way chop with the button allowing you to cycle between no chop, 2 way chop, and 3 way chop). 5. Finally, while the microcontroller is loaded with the signal for a 2-way chop, press and hold the ‘reset’ button.

	Using serial debugging, verify that the microcontroller returns to the unpressed value of the ‘chop’ button and that this value is the same even after releasing the reset button. 6. Organize all recorded values in a table
3. The pot count on the LCD Display should read ‘0’ upon the press, hold, and subsequent release of the ‘reset’ button in a second or less.	1. Put any number of chips through the contraption’s tunnel, such that the LCD display does not display a ‘0’ pot count value. Next, press and hold the ‘reset’ button. The value on the LCD display should read ‘0’, and should continue to read ‘0’ after the ‘reset’ button is released. The LCD display should update to ‘0’ pot count in a second or less of pressing the reset button, a stopwatch can be used to check this.
4. The LCD display should display the number of ways the pot is being chopped, and the respective color denominations. If the pot cannot be evenly chopped it will display the color denominations for the greater split, or the greater two splits. Along with the difference between this split and the least valued split. (If one half is one white chip more than the other, it will say some variation of ‘+1 White Chip’).	1. Put a number of chips through the contraption’s tunnel that cannot be divided by two or three evenly using the available chip values. 2. Press and Hold the ‘chop’ button. Ensure that the LCD display shows a 2 way chop, the color denominations for the greater half of the split, and the difference from the lower split in at most 2 seconds of the button press.

	3. Release the ‘chop’ button, ensuring that the LCD display shows the same information as the previous step. 4. Perform steps 2 and 3 two more times, ensuring that the LCD display updates to reflect that of a 3 way chop on the first iteration. On the second iteration, the display should show the same information as it did prior to the first iteration of step 2- just the pot count and the color denominations for 1 winner. 5. Perform 1 final iteration of steps 2 and 3 such that the LCD display shows the information for a 2-way chop once again. 6. Press and hold the ‘reset’ button, and ensure that the LCD display only displays the pot count of ‘0’ within at most 2 seconds of releasing the button.
5. The LCD display will display ‘COMM ERROR’ in at most 5 seconds if the microcontroller detects a connection failure.	1. Load a build of the microcontroller firmware where the connection failure is set to high. Ensure that the display shows an error message in at most 5 seconds of the firmware being loaded in.
6. The LCD display and the microcontroller should reflect expected changes in at most 2 seconds of the user hitting a button.	1. Insert any number of chips into the tunnel of the contraption. 2. Press and hold the ‘reset’ button,

	ensuring that the value on the display and the value read by the microcontroller using serial debugging is updated in at most 2 seconds of the user hitting pressing the button. 3. Add one more chip. 4. Press and release the ‘chop’ button, ensuring that the LCD display and the value read by the microcontroller using serial debugging is updated in at most 2 seconds.
7. The LCD display appends the pot count, according to the monetary value associated with the chip that is inserted.	1. Insert the maximum amount of chips the contraption supports, of varying colors. 2. Ensure that this display shows the right monetary value corresponding to each chip that gets organized, and that the final sum is the correct value that was inserted at the start. The correct pot count should be displayed within 2 seconds of the last chip being dispensed to its correct bin.

<u>Tolerance Analysis</u>

Part (Operating at 3.3V)	**Max Current Draw at 3.3V**	**Comments**
STM32F103C8T6 (MCU)	50.3 mA	External clock(2), all peripherals enabled @72 MHz, Ta = 105 ° C
TCS3200 (Color Sensor)	2 mA	Power-on mode
NHD-0216HZ-FSW-FBW-33V3C (LCD Display)	3 mA	Max Supply Current
Total Max Current Drawn	55.3 mA < 800 mA (Max Supply Current for LT1117-3.3V)	

Variable	**Value**	**Comment**
Tj	150 ° C	Maximum Junction Temperature for LT1117 Voltage Regulators
Iout	55.3 mA	Max Current Draw of Components @3.3V
Vin	9 V	9 V battery voltage
Vout	3.3 V	3.3 V to power above components
Θjc	15° C/W	Thermal Resistance, Junction to Case, for LT1117-3.3V

Θca	45° C/W	59 °C/W is the max Junction to Ambient thermal resistance listed on Data Sheet
Ta	30° C	Assuming warm board

Calculating Tj = 48.9126 ° C < 150 ° C. LT1117-3.3V Regulator will suffice.

Using 2x LM350 for each 6 V output. This will give us a 3 A max current output for each motor.

Part (Operating at 6V)	**Max Current Draw**	**Comments**
1x Servo Motor (Base)	600 mA	Claims to draw a maximum of 200mA (but we are going to allocate 600 mA just to be safe) current for a 6V Continuous Rotation Servo (#900-00008)
1x Linear Actuator (Pusher)	550 mA @6V	550 mA max current draw at a 6 V input for the PQ12-63-6-S Linear Actuator.
Total	600mA < 3 A (Base Motor) 550 mA < 3 A (Linear Actuator)	Max Current draw for both 6 V components are under the 3A threshold that the LM350 can output.

Variable	Value	Comments
Tj	125 ° C	Maximum Junction Temperature for LM350 Voltage Regulators
Iout	600 mA , 550mA	Max Current Draw of each Component (one for each regulator) @6V
Vin	9 V	9 V battery input voltage
Vout	6 V	6 V to power the above components.
Θjc	1.5° C/W	Thermal Resistance, Junction to Case, for LM350.
Θca	33.5 ° C/W	35 °C/W is the max Junction to Ambient thermal resistance listed on Data Sheet (NDS)
Ta	30° C	Assuming warm board

Calculating Tj = 87.75 ° C (Regulator for Linear Actuator), 93° C (Regulator for Motor) < 125 ° C. LM350 Regulator will suffice.

Cost and Schedule

Cost Analysis

Per the University of Illinois Urbana-Champaign's Website, the average electrical-engineer's starting salary is in the ballpark of $85,000 which can be divided to $40/hr. Therefore, $40/hr *2.5 * 60 * 3 Members = $18,000. According to the machine shop our project was quoted for 40 man hours at a rate of $56/hr, so the cost of the machine shop work would be $56/hr * 40 hrs = $2240. Next we can add the cost of parts to the cost of labor, and get a total of **$20,386.19**.

Description	Manufacturer	Quantity	Price	Link
LM350 Voltage Regulator	Digikey	2x	$2.02000 *2= $4.04	https://www.digikey.com/en/products/detail/texas-instruments/LM350T-NOPB/8901
LT1117CST#PBF-ND Voltage Regulator	Digikey	1x	$6.18000	https://www.digikey.com/en/products/detail/analog-devices-inc/LT1117CST-PBF/890204
Continuous Rotation Servo (#900-00008	Digikey	1x	$19.95	https://www.digikey.com/en/products/detail/parallax-inc/900-00008/1774454

PQ12-63-6-S Linear Actuator	Digikey	1x	$85.00	https://www.digikey.com/en/products/detail/actuonix-motion-devices-inc/PQ12-63-6-S/12317306
STM32F103C8T6 (MCU)	Digikey	1x	$6.42	https://www.digikey.com/en/products/detail/stmicroelectronics/STM32F103C8T6/1646338
TCS3200 (Color Sensor)	Digikey	1x	$7.90	https://www.digikey.com/en/products/detail/dfrobot/SEN0101/6588457
NHD-0216HZ-FSW-FBW-33V3C (LCD Display)	Digikey	1x	$13.00	https://www.digikey.com/en/products/detail/newhaven-display-intl/NHD-0216HZ-FSW-FBW-33V3C/2773591
10u Cap	Digikey	1x	$0.24000	https://www.digikey.com/en/products/detail/kemet/C322C104M5U5TA7301/3725993
22u Cap	Digikey	1x	$0.40000	https://www.digikey.com/en/product

				s/detail/cal-chip-electronics-inc/GMC32Z5U226Z16NT/14288400
9 V Battery	Digikey	1x	$2.460	https://www.digikey.com/en/products/detail/duracell-industrial-operations-inc/9V-MN1604/13280363
910 Ohm Resistor	Digikey	2x	$0.20	https://www.digikey.com/en/products/detail/yageo/CFR-25JB-52-910R/3749
240 Ohm Resistor	Digikey	2x	$0.20	https://www.digikey.com/en/products/detail/yageo/CFR-25JB-52-240R/1342
1 u Cap	Digikey	2x	$0.20	https://www.digikey.com/en/products/detail/kemet/C0402C105K9PAC7867/1090778
Total Cost of Materials			$146.19	

Schedule

We will assign our schedules to that of the course. The three group members will mostly work on the same tasks.

Week of	Justin	Marvin	Anish
9/25	Finish design document, finalize BOM, begin PCB design	Finish design document, finalize BOM, begin PCB design	Finish design document, finalize BOM, begin PCB design
10/2	Have design review, finalize CAD and machine shop conversations, begin calibrating components on breadboard	Have design review, finalize CAD and machine shop conversations, begin calibrating components on breadboard	Have design review, finalize CAD and machine shop conversations, begin calibrating components on breadboard
10/9	Finalize PCB design, order from PCBWay, complete teamwork evaluation	Finalize PCB design, order from PCBWay, complete teamwork evaluation	Finalize PCB design, order from PCBWay, complete teamwork evaluation
10/16	Second round of PCBWay orders, continue calibrating on breadboard	Second round of PCBWay orders, continue calibrating on breadboard	Second round of PCBWay orders, continue calibrating on breadboard
10/23	Complete individual progress report, continue breadboard calibration	Complete individual progress report, continue breadboard calibration	Complete individual progress report, continue breadboard calibration
10/30	Receive PCB, begin soldering and assembling design	Receive PCB, begin soldering and assembling design	Receive PCB, begin soldering and assembling design
11/6	Finish calibration and finish assembling design	Finish calibration and finish assembling design	Finish calibration and finish assembling design
11/13	Mock demo, prepare final presentation	Mock demo, prepare final presentation	Mock demo, prepare final presentation

11/20	Fall break - continue working on design and presentation	Fall break - continue working on design and presentation	Fall break - continue working on design and presentation
11/27	Final demo	Final demo	Final demo
12/4	Final presentation	Final presentation	Final presentation

Discussion of Ethics and Safety

Ethically, as a project that relates to money and the distribution of monetary equivalent chips, it is very important that we maintain an accurate count of chip value. Any error in the logic and sorting of the chips could result in an unfair financial loss to a player, which can compromise the entire game. The premise of our solution is to eliminate intentional and unintentional errors in home poker games, while increasing the efficiency of the game itself which adheres to Section I.1 of the IEEE code of ethics: "to hold paramount the safety, health, and welfare of the public"[1]. Since, our solution also attempts to eliminate manipulating pots, as described by our 4th high level requirement, it also supports Section I.4 of the IEEE code of ethics being "to avoid unlawful conduct in professional activities"[1]. This also means it is of the utmost importance that our microcontroller unit is not compromised to ensure the security of the game. Although it may not seem immediately relevant, our project will collect user data and data about the value of the pot. Even this seemingly harmless data can raise privacy concerns, so it is essential to adhere to IEEE and ACM guidelines concerning data privacy. As students, it would be difficult to code our microcontroller in a way that makes it immune to hacking or interference. The best we can do to combat interference is by making our logic as simple as possible. Since our solution targets the home-game setting, we can say almost definitely that we will be able to provide a solution that is "usably secure" per ACM ethics guideline 2.9.

From a safety standpoint, any mechanism that uses motors and electrical components presents a safety hazard. Our device will feature a solid enclosure around the motors and moving parts in order to prevent any hair, jewelry, or loose clothing items from getting stuck in the motors and causing harm. In addition to this, our design will feature insulation around any wires and loose electrical components to prevent any harmful contact.

It is also important to abide by the strict IEEE and ACM guidelines against plagiarism [3]. Although there are a number of chip sorting mechanisms available on the market today, none of them are directly targeted for home games. This is reflected in the cost of the device. As our product features proprietary hardware and software - targeting a brand new demographic - we can safely avoid any plagiarism.

Finally, any product that supplements gambling is subject to review from a number of gambling governing bodies. The Illinois Gaming Board is the governing body in the state of Illinois, and controls a regulatory and tax collection for gaming in the state. Although regulations will be less stringent due to the home game target demographic, any gambling product will be subject to review before being commercially available to the public.

Citations

[1] "IEEE code of Ethics," IEEE, https://www.ieee.org/about/corporate/governance/p7-8.html (accessed Sep. 14, 2023).

[2] R. Fee, "6 reasons why live poker is easier than online poker," Upswing Poker, https://upswingpoker.com/live-poker-vs-online-poker-easier/ (accessed Sep. 14, 2023).

[3] Zion Market Research, "Trading card game market size, share and demand 2030," Zion Market Research, https://www.zionmarketresearch.com/report/trading-card-game-market (accessed Sep. 14, 2023).

Datasheets:
https://www.ti.com/lit/ds/symlink/l293.pdf