Wheelin and Dealin

By

Soham Makani

Leo Chen

Design Document for ECE445
Senior Design
Fall 2021

TA: YiChi Zhang

23 September, 2021

1 Introduction

1.1 Background

The first playing cards appeared in Europe in the late 1300s and early 1400s, however there is evidence of older cards existing in China, India, Korea, Persia, and Egypt. Cards were used for a variety of purposes, primarily entertainment and gaming. The origin of our problem comes from Texas hold'em, a gambling game invented in the early 1920s. A game of Texas hold'em rarely involves a single hand and generally games last multiple hours with hundreds of hands being played. [6] As each hand requires a freshly shuffled deck a player or designated dealer must shuffle and deal the cards. This is an incredibly tedious process and shuffling and dealing can make up for more than 10% of game time. This also often takes one player out of the game as a player will need to shuffle and deal cards, for such a repetitive process why is there no automation?

This problem presents itself in the majority of card games beyond just Texas hold'em. Games like president and hearts require the entire deck to be randomized and dealt each time a new game starts, and popular chinese gambling game Zhao Peng You requires two decks to be randomized and dealt. This process is repetitive and tedious putting a thorn in an otherwise enjoyable experience. For something that is so repetitive we have a solution to automatically shuffle and deal cards.

1.2 Solution

Currently there doesn't exist a cheap programmable card dealer for a commercial audience. We propose a machine that is able to shuffle and deal hands automatically without need for player interaction and intervention. The solution would be priced for recreational use, table mounted, and suitable for a variety of different sized tables and player configurations. For the sake of consistency and to limit the scope of our project we will confine the table surface to be felt, and the only game available to be Texas Hold'em. Our solution will eliminate the need for humans to shuffle and deal cards, as well as not fall significantly behind the efficiency of a human dealer.

1.3 Visual Aid

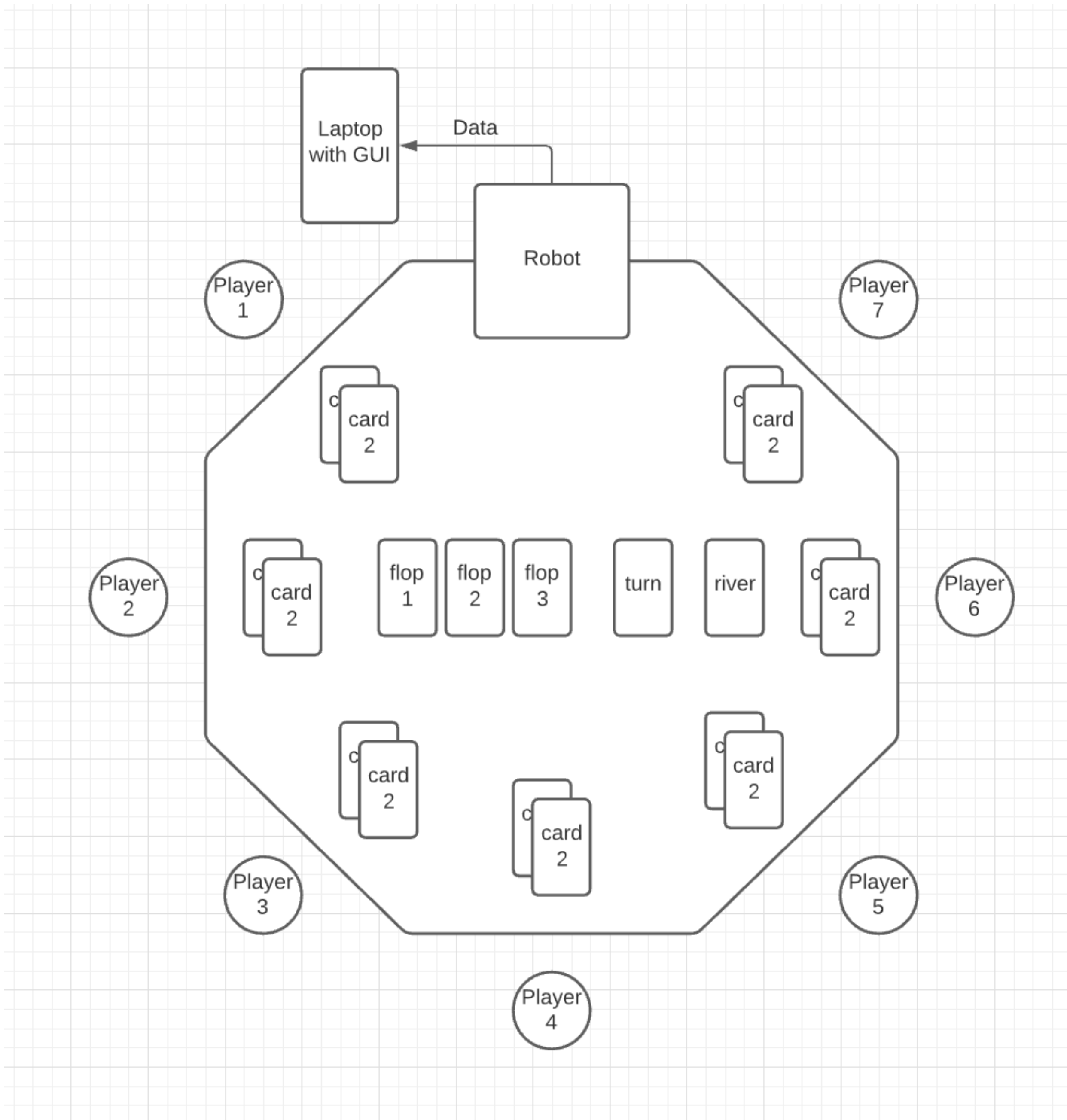

Figure 1: Visual Aid

1.4 High-Level Requirements List

i. The dealer should eliminate the need for a player or designated dealer to shuffle and deal a single Texas Hold'em hand. Furthermore the mechanical dealer should be efficient enough to account for less than 20% of total game time.

ii. When the dealer pitches a card to a player at least 90% of the card should land in a 30cm by 30cm square in front of the player face down 95% of the time. If a card lands facedown the trial will be considered a failure, if over 10% of the card lands outside the designated area the trail will be considered a failure. If both occur the trial will be considered a failure.

iii. The mechanical dealer should correctly identify the card being dealt over 95% of the time in under 500ms. The database should store at least 10000 cards worth of data and current game data should be displayed on a simple mobile app GUI with a latency of under 3.5 seconds.

2 Design

2.1 Block Diagram

The block diagram outlines 4 different portions that will work together in order to fulfill the high level requirements. The power subsystem consists of an AC/DC converter that can supply 5V DC power to all our components. The dealer subsystem is the section that removes a single card off the deck and deals it to the player. The system consists of 3 motors. One which removes a single card off the deck and gives time for a camera to scan the card, another that actually launches the card to the player, and one that is used to rotate the entire platform. The control system consists of the Atmega328 microcontroller that controls the sensors, and a raspberry pi that processes the images of cards to identify them and sends the data to an external database. The player detection system uses an ultrasonic sensor and a Pixy CMUcam image sensor. This is an arduino compatible sensor that has built in block detection and communicates over SPI. The ultrasonic sensor is used to calculate the distance between the machine and a player after the PixyCam identifies them. Finally, the card detection system consists of another simple camera, and the database and GUI to keep track of the cards dealt to each player. The photo-resistor will detect when a card should be scanned and communicate with the Raspberry Pi. The basic camera relays images to the Raspberry Pi, which then processes the data and sends it to an external database. This data is then displayed on the GUI.

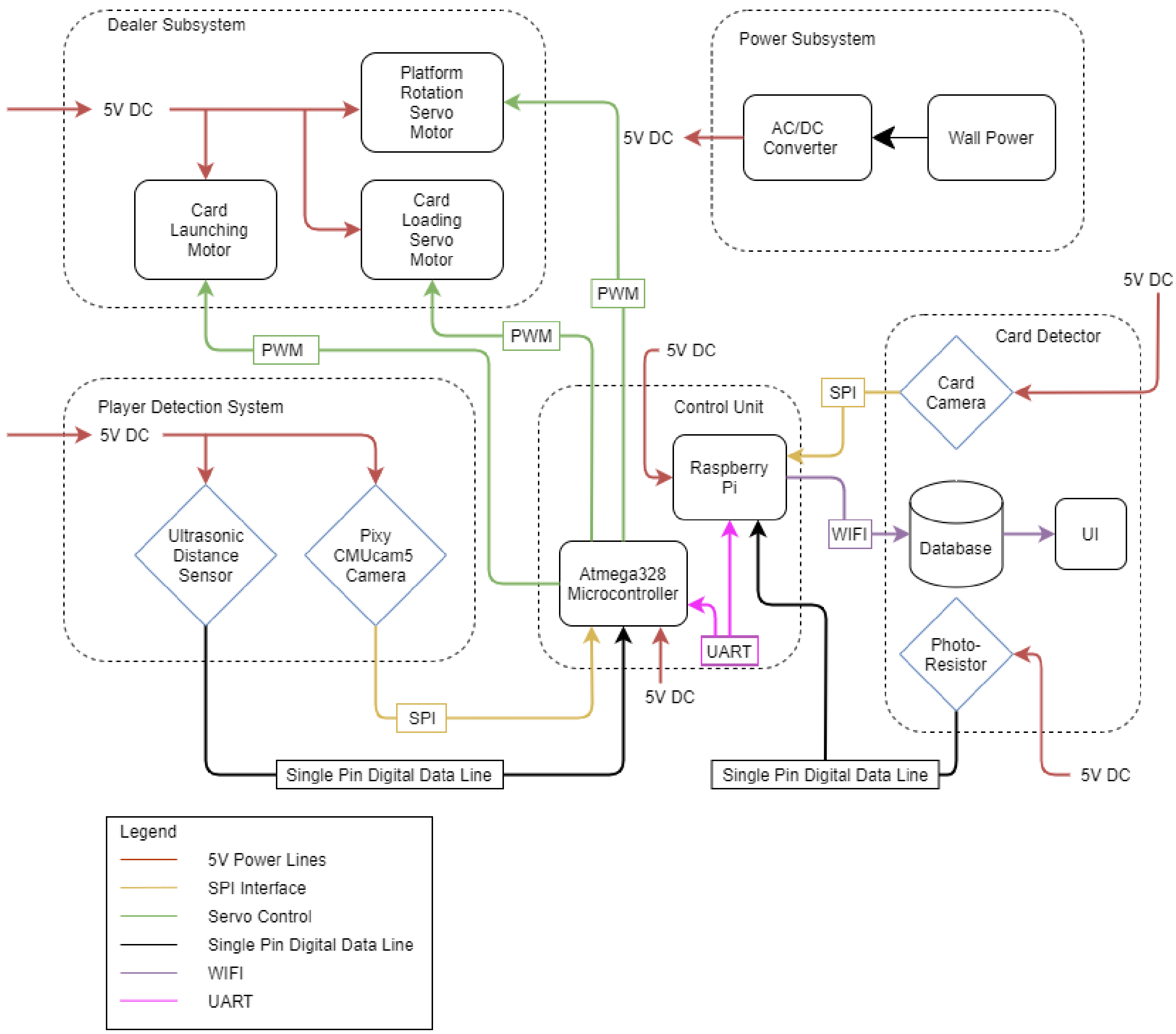

Figure 2: Block diagram of our system

2.2 Player Recognition Unit

This unit is responsible for identifying players and then identifying distance to players so we can accurately deal a card based on player distance. The unit is composed of two different sensors which both communicate with the control unit. The Pixy cam is responsible for identifying whether a player is sitting at the table and the Ultrasonic sensor is responsible for identifying distance to the player. There should not be any obstacles between the Pixy cam and the player nor the ultrasonic sensor and the player. If the Pixy cam is not able to identify the player we will place a 8.5’’x11’’ rectangular sheet in front of the player in order to identify whether a player is present.

2.2.1 Ultrasonic Sensor

The HC-SR04 is a simple compact ultrasonic sensor that's able to measure distance to the object in front of the sensor. To trigger the ranging we will send a single 10us pulse from our microcontroller. The module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The echo is a distance object that is pulse width and the range in proportion. We will calculate the range through the time interval between sending trigger signal and receiving echo signal using the formula: uS / 58 = centimeters away.[7]

Requirements	Verification
1. This sensor will identify players sitting 30cm away, 60cm away, and 90cm away with an accuracy of ± 5cm.	1. a. Sit a single player 30, 60, and 90 cm away. b. Trigger sensor to start ranging c. After receiving echo pulse width calculate location of player d. Verify that the calculated location is in line with tolerance of 30, 60, 90cm ± 5cm

2.2.2 Pixy Camera

The Pixy camera is a unique camera that has built in image and color recognition capabilities giving us the ability to quickly identify a player without external OCR which will take too long. The Pixy camera has many capabilities that we will explore once the camera arrives. The camera however must consistently identify that a player is sitting at the table and quickly send a signal to the microcontroller.

Requirements	Verification
1. The camera can detect a human a distance of 30, 60, 90 cm away and send a signal to the microcontroller within 200ms.	1. a. Program the sensor for the appropriate person or object b. Place a person or object the 30, 60, 90 cm from the sensor c. Cover the sensor with paper to prevent it from detecting anything d. Remove paper and time how long it takes for the microcontroller to receive a signal showing that the camera has detected a human. e. Verify that time taken to detect human is under 200ms

2.3 Card Dealing Unit

This Unit involves three motors that are each controlled by the microcontroller. The first motor is responsible for motion on the X axis rotating the entire dealer 360 degrees. The second motor slides a single card out from the deck to be dealt. The third motor shoots the card that has been slid out towards the player. Each motor or servo is controlled by a PWM signal sent by ATmega328 microcontroller

Requirements	Verification
1. The dealer must successfully deal cards to players 1,2,3 and 4 feet away face down into a 30cm by 30cm square with a success rate of over 95%. Trials with 10% of the card landing outside of the square will be considered a failure.	1. a. Adjust servo to player position based on angle. b. Start the card slider motor to slide the card. c. Start card dealer motor to deal cards to players. d. Verify 90% of the card is in a 30cm by 30cm predetermined square in front of the player. e. Repeat 100 trials where any trial with 10% of the card landing outside of the square considered a failure. If multiple cards are dealt the trial will be rerun.
2. The dealer will dispatch one card at a time to a player with a success rate of over 90%.	2. a. Adjust servo to player position based on angle. b. Start the card slider motor to slide card c. Start card dealer motor to deal card to player d. Verify that a single card is dealt every time, e. Repeat 100 trials where an event with multiple cards

	being dealt is a fail. Make sure we have at least a 95% success rate.

2.3.1 Card Slider Motor

This motor slides a single card away from the deck so it can be dealt by the card shooting motor. This motor will be positioned under the deck of cards and use a rubber wheel with high friction to slide a single card from the bottom of the deck. This motor requires high precision and low RPM so we have selected the SER0056 servo motor. Servo motors are position based and should support very high accuracy, allowing us to decide how far the card will be slid before being launched.

Requirements	Verification
1. The motor must be able to receive output from the microcontroller and stop in the desired angle accurate to ± 2 degrees.	1. a. Connect the motor to 5v DC power. b. Connect the motor to PWM out on microcontroller. c. Program Microcontroller to send variable angle settings to the motor. d. Verify the motor reaches the desired angle ± 3 degrees using a protractor.
2. motor slider system must be able to slide a single card 2 ± .1 inches forward and stop the card before it reaches the launcher.	1. a. Connect the motor to 5v DC power. b. Connect the motor to PWM out on Microcontroller. c. Verify that a single card is slid 2 ± .1 inches.

2.3.2 Card Shooter Motor

This Motor will launch the card after it has been slid out and scanned by the card slide motor. It needs to launch the card with accuracy and distance as well as face down. It will receive a motor speed from the microcontroller and then activate to launch the card to a player. This will be the most challenging portion of our project, we need to ensure that the card launched always ends up within a 12 by 12 square. The values we selected are based upon kinematic principles.

Requirements	Verification
1. The Motor should be able to rotate at 300 RPM, 600 RPM, and 900 RPM ± 10 RPM.	1. Connect the motor to 5v DC power a. Connect motor PWM pin to Microcontroller. b. Program Microcontroller to send variable RPMs to the motor. c. Verify motor spins at proper RPM using a mobile application or Tachometer.

2.3.3 Dealer Rotation Servo

This motor carries the weight of the entire dealer and rotates the dealer accurately and quickly. The SER0035 from DFRobot has a dynamic torque of 15.1kg.cm which is well within our weight tolerance and it has a no load speed of 60°/0.16s which should be much faster than desired for our dealer. The Servo takes a PPM signal which gives the servo position. We will need to vary the voltage delivered to the servo in order to ensure the servo speed is not too fast nor too slow.

Requirements	Verification
1. The servo must be able to rotate the dealer platform (~5lbs) 180 degrees within two seconds.	1. a. Connect the servo to DC power.

	b. Connect servo PPM pin to Microcontroller. c. Program Microcontroller to rotate servo 180 degrees. d. Time servo speed from microcontroller signal to full stop using stopwatch. e. Verify time is within two seconds.
2. The servo position after motor signal must be accurate to ± 3 degrees from calculations.	2. a. Connect servo PPM pin to Microcontroller. b. Program Microcontroller to rotate servo 180 degrees. c. Verify servo position upon stop using protractor.

2.4 Card Identifier Unit

This unit should operate largely independent from the main dealer unit. Although physically mounted on the dealer, there should be little to no interaction between the microcontroller that controls the main dealer unit and raspberry pi that operates the card identification. This unit should consist of a simple camera to capture images of the cards as they are slid from the deck before launch. The digital card data will be sent to a raspberry PI and then wirelessly sent to a server where OCR will be done on the image. After the card value and card suit is identified, the card data will be stored in a SQL database, where we can easily fetch data to be displayed in a mobile application.

Requirements	Verification
1. The GUI should receive and display the correct card and player information 95% of the time.	1. a. Boot Mobile app on spectator phone. b. Have bot deal 1 hand of Texas Hold’ em.

	c. Count number of cards correctly dealt, count number of cards correctly displayed. Mismatches are considered failures. d. Repeat for 20 hands and ensure 95% success rate, where incorrect cards are considered failed trials.

2.4.1 Camera

The camera will be a simple camera with low focal length so we can take images of cards close up. The camera outputs continuous digital data to the raspberry pi. In order to isolate the frames we want we will place a light sensor adjacent to the camera. When the light sensor is blocked by the card sliding over it, we will know which frame to isolate and perform OCR on. That frame will then be sent to a server for image processing.

Requirements	Verification
1. The camera must output image data to the Raspberry Pi, and provide a focused image suitable for OCR 95% of the time. The images should have a resolution of at least 1600 x 1200.	1. a. Connect the camera to the Raspberry Pi over the serial interface, and make sure the Pi receives image data. b. Test the camera at 1 inch focal length. c. Save images from Raspberry Pi to a separate desktop computer and confirm image resolution of 1600 by 1200 pixels. d. Test images with Tesseract OCR package. e. Verify that Tesseract is able to identify key points on the image 95% of the time.

2.4.2 Photoresistor Module

The photoresistor module contains a light sensitive resistor that changes resistance as the light intensity changes. The module has a single digital pin that will output low or high depending on the light intensity of the sensor. The module also contains a potentiometer that can be used to adjust the light threshold for the digital output.

Requirements	Verification
2. The photoresistor should output a digital signal to the Raspberry Pi. The signal should output a 0 when the sensor is given ambient lighting, and 1 when the sensor has a card covering it.	2. a. Connect the module to ground and 5V. b. Then we can test the digital pin out to see if we receive a signal. c. Connect the module to the Raspberry Pi. d. Test if the Pi receives the digital output. e. Place the module where it will be held on the machine. Modify the potentiometer threshold to read 0 when no card is in front of it and 1 when a card is over it. f. Place the module where it will be held on the machine and connect it to the Raspberry Pi. g. Verify the digital signal corresponds to whether a card is in front of the sensor or not.

2.4.3 Raspberry Pi OCR

The Raspberry Pi can run popular open source OCR libraries such as tesseract. This is what we will use to identify the cards. The model we use must be able to correctly identify the card type, card value, and card suit. This data will then be transferred to the database.

Requirements	Verification
1. The Raspberry Pi must be able to correctly identify the card type, card value, and card suit 95% ± 1% of the time. Misidentifying any of these attributes will be counted as incorrect, as well as not getting any result. The Pi must also get its identification within 500.	2. a. First, test tesseract on pictures of playing cards taken from the card scanning camera. b. The code will run on an external PC. c. Verify we can hit the accuracy value of greater than 95%. d. Next upload the tesseract code onto the Pi, and test it with saved pictures of cards from the card scanning camera. e. Verify the program is able to identify the card correctly over 95% of the time and runs within the time limit of 500 ms. f. Finally test the OCR code with live data coming in from the card scanning camera. g. Verify we have an accuracy of over 95% and our code runs within the time limit of 500ms.

2.4.4 Database

Because we will do OCR on the Raspberry PI, we will only store data for card type, card value, and card suit. Card type will be who or where the card was dealt to, this will be supplied by the microcontroller, the card suit and card value will be identified through OCR. The database will be a postgreSQL database.

Requirements	Verification
1. The Database must be able to be updated by the Raspberry Pi. No data should be lost, and new data should be received within 3 seconds of being processed by Raspberry Pi.	1. a. Create a PostgreSQL database and create a table to hold game data. b. Manually insert data using queries to verify database functionality. c. Insert dummy data from Raspberry Pi to verify that the Pi can upload data to the database with 0 errors. d. Upload live data from the raspberry Pi and time the time it takes for data to appear in the database. Verify that the data is accurate, no data has been lost, and time for upload is under 3 seconds.

2.4.5 Web Application

The mobile application will be a simple GUI that will let users see the current hand being played. The mobile application will be written with Node.js + React. The GUI will be simple and easy to understand.

Requirements	Verification
1. The mobile app must be able to connect to a server using WIFI to access correct data from the database.	1. a. Start back end server. b. Fetch data using simple ajax requests. c. Log data to console. d. Verify that data in the database is consistent with data being displayed in the web console.
2. The mobile app must be able to display the data fetched from PostgreSQL database where displayed data is 100% consistent with data in the database.	2. a. Start the mobile app and server. b. Fetch data using simple ajax requests. c. Log data to console . d. Verify data being displayed is identical to the data in the database.

2.5 Power

This Unit powers the entire dealer, Because the dealer should run for extended periods of time, we expect the dealer to be plugged into a power supply plugged into wall power.

Requirements	Verification
1. The power supply must supply a constant 5V ± .25V to all of the machine's components.	1. a. Plug the power supply into wall power, and test output with a multimeter to make sure we are in the range 5V ± .25V. b. Plug the power supply into our circuit board and make sure all components have power with a multimeter c. Plug power supply into Raspberry Pi and ensure it boots and functions.

2.6 Control Unit

The control unit will feature a ATMEGA328-AU microcontroller which will be able to control all three of our motors as well as take inputs from our sensors. The microcontroller will receive one digital and one analog input. The microcontroller will send two PWM signals as well as communicate with a stepper motor driver to maneuver the dealer and deal cards. The Microcontroller will be programmed to deal a single hand of Texas hold’em and can be reset with a single button to start dealing another hand.

Requirements	Verification
1. a. The microcontroller can send signals to and control the rotation servo. b. The microcontroller can send signals and control the card launcher motor. c. The microcontroller can send signals and control the card slider servo.	1. a. Connect motors to appropriate pins on PCB. b. Send PWM signals to servo and motor. c. Verify that the motor spins at desired speed and the servo rotates to the appropriate position.
2. The microcontroller can receive and process digital output from the Ultrasonic distance sensor.	2. a. Connect the Ultrasonic sensor to appropriate pins on PCB. b. Program microcontroller to take echo signal from ultrasonic sensor and calculate distance. c. Send a signal from the microcontroller to start ranging. d. Store results from calculations on the board. microcontroller memory e. Verify correct distance by examining memory.
3. The microcontroller can receive digital output from both Pixy cameras.	3. a. Connect Pixy camera to appropriate pins on PCB. b. Program Pixy camera to identify shape or player. c. Connect LED to microcontroller and program microcontroller to light LED once it receives signal from Pixy camera. d. Place shape or player in

	front of the pixy camera. e. Verify LED only lights up when the player or shape is first placed in front of the Pixy camera.

2.7 Schematic

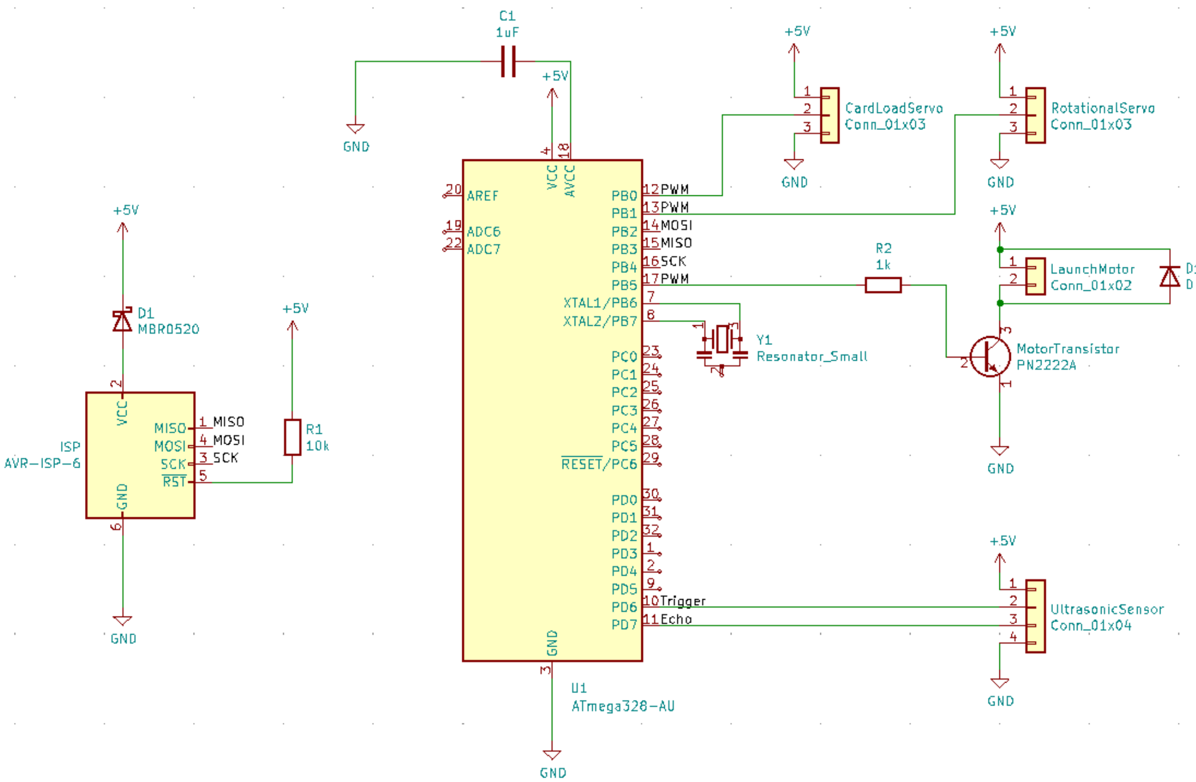

Figure 3: Microcontroller Schematic

2.8 Tolerance Analysis

2.8.1 Card Shooter Physics

2.8.1.1 Physics of a Card Sliding Across the Table Surface

As the card slides across the table surface, it will be slowed down by friction and eventually stop. We need to calculate the kinetic energy the card should have in order to travel the distances we want, which are 30, 60, and 90 cm.

When we first launch the card, it will be above the ground and slowly falling. Because the card is so low to the ground, we will ignore the air resistance and assume it accelerates downwards at 9.8 m/s^2. We can also use the dimensions of a playing card (64 x 89 x 0.17 mm) and the fact that they have a weight of 280 g/sqaure meter to find a mass of 1.4 grams per card. The height of a card is negligible for its weight. We can say that the card will have an initial kinetic energy of K, with an initial velocity of V, where

$$V = \sqrt{2K/.28}$$

First, the cards will be launched from a height of 10 cm off the ground. Using the equation:

$$T = \sqrt{2d/a}$$

Where T is time, a is acceleration, and d is distance. After accelerating downwards at 9.8 m/s^2, it will touch the ground after .14 seconds. In this time it will have traveled a distance of .14V.

Now that it is sliding across the surface, we should take friction into account. The closest we could get to the dynamic coefficient of friction for a playing card on a table was .2, from https://www.engineeringtoolbox.com/ .

The force of friction on a sliding object is:

$$F = \mu_k N$$

Where N is the normal force equal to m*g and μ_k is the coefficient of dynamic friction. Calculating the force of friction on the card gives us a force of 0.0027N and a deceleration of 1.96 m/s^2. The time it takes for the card to stop can be calculated from the equation:

$$T = v/a.$$

In our case where we use our initial velocity V, the time taken is V/1.96 seconds. Finally, we plug our values into the equation:

$$s = vT + \frac{1}{2} aT^2$$

We also need to combine this distance with the distance we calculated for the card flying through the air. We now have an equation we can use to calculate the initial card velocity for different distances.

This equation is:

$$d = .14V + V \times (V/1.96) + \frac{1}{2} \times 1.96 \times (V/1.96)^2$$

We insert our original distance values of 30, 60, and 90 cm for s, and the values we calculated earlier to find different values of V, and the respective values for Kinetic Energy. These values are shown in the table below.

Distance (cm)	Velocity (m/s)	Kinetic Energy (J)
30	.54	.00020
60	.80	.00044
90	.99	.00069

Table 2.8.1: Velocity and Kinetic Energy values

These numbers may be slightly inaccurate, since we could not find an exact coefficient of sliding friction. However, they will be useful in finding base rpm values for our motors when we start testing.

2.8.1.2 Physics of Launching the Card with a Spinning Wheel

The rotational kinetic energy of a spinning object is given by the equation:

$$KE_{Rot} = ½I\omega^2$$

Where ω is the angular velocity and I is the moment of inertia. The moment of inertia for a solid spinning wheel is given by the equation:

$$I = ½mR^2$$

Where m is the mass and R is the radius of the wheel.

In our case, we can ignore the friction between the launching wheel and the card since there shouldn't be any slipping. This means that all of the rotational kinetic energy of the wheel can be converted into translational kinetic energy for the card. Using the previous calculations for the kinetic energy needed to slide the card different distances, we can decide on the RPM for our motor to spin at.

Angular velocity can be calculated from rpm using the conversion:

$$1\, rpm \ = \ 30/\boldsymbol{\pi}\, rad/s$$

Our wheel has a mass of 25 grams, and a radius of 5cm. We use this information to calculate the moment of inertia using the formula:

$$I \ = \ ½mR^2$$

This gives us a moment of inertia of $3.125*10^{-5}$ kGm^2 .

We can insert this value into the equation for kinetic energy and solve for our angular velocity with the following formula:

$$\omega \ = \sqrt{2KE/I}$$

Using the values for kinetic energy from section 2.8.1.1 we can solve for the motor rpm needed to launch the card at different distances. This gives the table of values for angular velocity and rpm below.

Distance (cm)	Kinetic Energy (J)	Angular Velocity (rad/s)	RPM (rotations/min)
30	.00020	3.58	34.19
60	.00044	5.31	50.70
90	.00069	6.65	63.50

Table 2.8.2: Angular Velocity and RPM

2.8.1.3 Physics of Sliding One Card Off The Deck

The final problem we have to deal with is sliding a single card off the bottom of the deck with a motor and wheel. In this case, we have to overcome the static friction between two cards, and tune our motor to provide the right amount of force.

The force of friction between the cards will depend on the size of the deck, since we are removing cards off the bottom. However, since the deck is held in place, we will do calculations for when the deck is at its largest since extra force should not cause any issues.

The Normal force on the card at the bottom will be equal to the force exerted by all the cards above it. We can use the following equation to calculate the value.

$$F = 9.8 \times 51 \times g$$

Where G is the mass of a single playing card, and 9.8 is the acceleration due to gravity. We get the value of the normal force to be 0.7 Newtons.

Next we take a conservative estimate of the coefficient of static friction at .3 between the surfaces of the cards. Using the equation:

$$F = \mu N$$

We get that the force of friction between the two cards is .21 N.

Finally, we can calculate the torque needed to move our wheel and remove the card from the deck. For this task we are using a wheel with a radius of 2 cm. Using the equation:

$$\tau = r \times F$$

We can calculate that the necessary torque is .0042 Newton-meters, which is well within the range of our motor. Additionally, excess torque should not be an issue, since the wheel only touches the edge of the deck, and will not remove more than one card.

3 Cost and Schedule

3.1 Cost

3.1.1 Cost of Parts

Part	Description	Qty	Cost
HC-SR04	Ultrasonic sensor for distance	1	\$3.95
ATmega328P	Microcontroller	1	\$4.25
DFRobot SER0035	Servo to rotate dealer	1	\$18.05
DFRobot SER0037	Servo to slide card	1	\$7.9

PPN7PA12C1	Motor to Shoot the card	1	$3.22
Pixy Camera	Camera to detect player	1	$59.95
Photoresistor Module	Light sensor to check if card is present	1	$3.55
Total:			$100.87

Table 3.1.1: Cost analysis

3.1.2 Cost of Labor

According to UIUC ECE statistics, the average salary of a computer engineering graduate is $96,992 per year. Assuming a standard 40 hour work week and a 52 week year, the hourly salary comes out to ~$47/hr. We have two Computer Engineers roughly 10 hours each per week. This comes out to $940 per week for 9 weeks. The total cost of labor will be **$8460.00**.

3.2 Schedule

Week	Soham	Leo
10/04	Complete CV for cards on PC. Design Review	Complete and Finalize PCB version 1 and complete order. Design Review
10/11	Complete database and UI.	Calibrate Pixycam onboard IC to identify players around a table.
10/18	Link Raspberry Pi to database. Test photoresistor and verify thresholds for detecting a card. Test CV code for cards on Raspberry Pi and verify it	Start testing mechanical systems using PCB. Verify values for motor spin speeds. Verify that the ultrasonic sensor and Pixycam interact with the microcontroller properly.

	identifies cards correctly according to our guidelines.	
10/25	Link Raspberry Pi system to Arduino system. Start initial trials of simple hand dealing.	Complete version 2 of PCB and place order. Finalize mechanical card dealing system and player detection system
11/01	Start testing and ironing out bugs from linking systems. Verify that the machine is able to deal 1 card to all detected players, with correct identification. Verify the card correctly appears in the database and GUI.	Start testing and ironing out bugs from linking systems. Verify that the machine is able to deal 1 card to all detected players, with correct identification. Verify the card correctly appears in the database and GUI.
11/08	Modify the working system to deal correctly for Texas-Hold'em.	Modify the working system to deal correctly for Texas-Hold'em.
11/15	Fix any bugs that appear and prepare a mock demo.	Fix any bugs that appear and prepare a mock demo.
11/22	Fall break	Fall break
11/29	Prepare for demonstration and Mock Final presentation.	Prepare for demonstration and Mock Final presentation.
11/26	Prepare Final Presentation and Report	Prepare Final Presentation and Report

Table 3.2.1: Schedule

4 Ethics and Safety

According to the ACM code of ethics we should aim to “Contribute to society and to human well-being” [2]. The device aims to better the experience of all those who play Texas Hold’em games. The device can be extended to improve the experience of all card game players and make it simpler and more efficient to play Texas Hold’em.

The IEEE code of ethics “holds paramount the safety, health, and welfare of the public” [2] We aim to make this device as safe as possible. A few safety concerns such as fast spinning motors will be addressed using an enclosure to house the entire system making it nearly impossible to interact with and access internal parts. As the device is quite complex and involves many moving parts we will provide a detailed instruction manual stating how to load cards, operate the machine, and safety considerations when using and operating the card dealer.

As we will need to do considerable soldering we will adhere to all soldering guidelines and instructions supplied under the University of Cambridge department of engineering[5]

Beyond safety considerations because we are making a machine that deals a gambling game we will adhere to the Illinois Gaming Board. The Illinois Gambling act states “For the examination of all mechanical, electromechanical, or electronic table games, slot machines, slot accounting systems, sports wagering systems, and other electronic gaming equipment, and the field inspection of such systems, games, and machines, for compliance with this Act, the Board shall utilize the services of independent outside testing laboratories that have been accredited in accordance with ISO/IEC 17025 by an accreditation body that is a signatory to the International Laboratory Accreditation Cooperation Mutual Recognition Agreement signifying they are qualified to perform such examinations.”[4] We will not be supplying casinos with this card dealer and anybody trying to use our dealer for non recreational gaming should adhere to all Gaming Board regulations. Similarly we will not run games that take rake nor receive income from games as it is illegal under the Illinois gambling act to derive financial gain from games without a gaming license [4].

5 References

[1] D. Gotterbarn. ACM Code Of Ethics And Professional Conduct. (2020). Accessed: Sept 29, 2020. [Online]. Available: https://www.acm.org/code-of-ethics

[2] IEEE. IEEE Code Of Ethics. (2020). Accessed: Sept 29, 2020. [Online]. Available:https://www.ieee.org/about/corporate/governance/p7-8.html

[3] "Friction and friction coefficients," *Engineering ToolBox*. [Online]. Available: https://www.engineeringtoolbox.com/friction-coefficients-d_778.html. [Accessed: 01-Oct-2021].

[4] "GAMING (230 ILCS 10/) Illinois Gambling Act.," 28-Jun-2019. [Online]. Available: https://www.ilga.gov/legislation/ilcs/ilcs3.asp?ActID=1399&ChapAct=230%C2%A0ILCS%C2%A010%2F&ChapterID=25. [Accessed: 01-Oct-2021].

[5] I. Slack, "Soldering safety," *University of Cambridge*, 27-Mar-2018. [Online]. Available: https://safety.eng.cam.ac.uk/safe-working/copy_of_soldering-safety. [Accessed: 01-Oct-2021].

[6] "Omaha," *Encyclopædia Britannica*. [Online]. Available: https://www.britannica.com/topic/poker-card-game/Omaha. [Accessed: 01-Oct-2021].

[7] "Ultrasonic ranging module HC - SR04 - SparkFun Electronics." [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf. [Accessed: 01-Oct-2021].