ECE 4950: Project 4 Report

Team 21

Andrew Berkeland, Andrew (AJ) Eubanks, Steven Johnson, Eric Mitchell, Alvi Topuzi

12/09/2024

Executive Summary

The goal of our project was to design a robot that can play ‘Connect Four” against a user in real time. Humans are not allowed to assist the robot during operation. The user will select one of two colors and decide who plays first: the robot or the user. The game ends with a win, which occurs when four same-color pieces are in a straight line. Through working on this system, we worked on our image recognition, programming, and 3D design skills. We faced some challenges on the way, such as getting the electromagnet to move the pieces through the gameboard. We also had to design and construct the base for the game board, practicing engineering-for-manufacturing, along with other engineering scenarios.

Engineering Requirements

- The robot plays the game according to the rules of Connect 4
- Uses the provided game stage and electrical and mechanical interfaces
- Costs < $300*
- Use the provided Arduino, camera, and DC motor.
- Uses a power source for custom electronics (beyond the standard computers and peripherals such as the USB camera).
- No batteries
- Reliable, Durable, Safe
- Low noise – no hearing protection required
- Guarding as necessary to protect users
- Fast solving times
- Easy to use/user-friendly
- Electric/electronic circuits built from off-the-shelf components.
- Runs autonomously – user places their own pieces
- Must have at least one degree-of-freedom that has closed loop position control using the provided DC motor where the loop is closed through the Arduino.
- GUI must be available which displays the initial configuration of your game set up and the final configuration as specified by the user
- Systems will have to play a single game of Connect 4 with the goal of getting 4 Orange/Purple coins in a row
- Reliable Suppliers

System Architecture

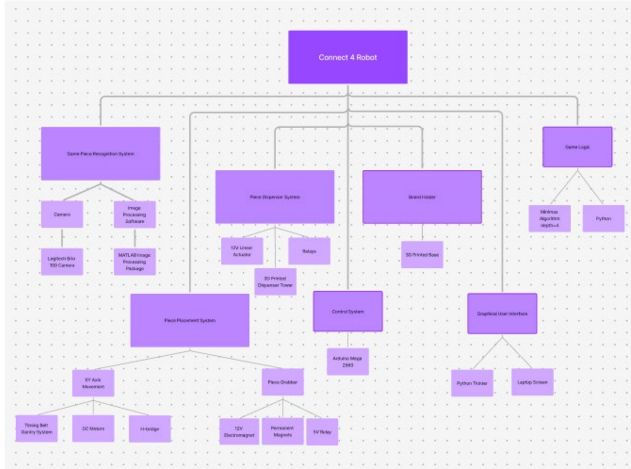

Figure 1: Final project functional decomposition

In this project, we were given many components that we had to include in our final design (see Figure 1). Beginning with the Logitech Brio 100 camera, we needed a way to take pictures with the camera and use them in software. We chose to use the MATLAB image acquisition and image processing toolboxes to bridge our hardware and software. Combining both the camera and MATLAB, we were able to create image processing software that could recognize where new game pieces were placed.

We were also provided with a DC motor equipped with a hall effect sensor encoder. As a project requirement, we had to include the motor in one of our axis movement systems. We decided to go with an X and Y axis movement system. Using timing belts, pulleys, and a gantry system, we created a way to move in two directions. We still needed to find a way to collect the game pieces. By attaching the provided 12V electromagnet, we now had a way to grab pieces. To effectively use the electromagnet, we equipped our game pieces with permanent magnets. The electromagnet requires 12V DC power. Since we had to use the Arduino Mega 2560 board, we only had access to 5V switching logic. We could not control the electromagnet with the Arduino directly. To fix the issue, we used a relay to convert the 5V switching logic of the Arduino to 12V DC switching for the electromagnet.

To dispense game pieces for the robot, we decided to use a 12V linear actuator. We did not use the provided servo motor because it was imprecise and could not easily overcome the magnetic force between pieces. It was also not a required component. We 3D printed a dispenser tower to hold our pieces. Since the actuator is a 12V device and we needed bi-directional movement, we needed two relays to control it.

The system frame holds all the components together. We opted to use PLA filament instead of wood or 80/20 aluminum. The PLA filament was more readily available than either of the other options. It was also lightweight compared to wood. Lastly, it was easier to modify than 8020 aluminum pieces.

Another game requirement is that we needed a graphical user interface. Since we had a team member who is experienced in Python programming, we chose to go with Tkinter as our visual library. It provides enough visual aspects to create a clear and aesthetically pleasing GUI design.

Lastly, we needed software to control our game logic. Similar to the user interface section, we chose Python due to team member experience. For our winning algorithm, we used the Minimax algorithm. The algorithm was readily available. Also, it maximized the computer's chance of winning by thinking of moves four steps in advance.

Electronic Components

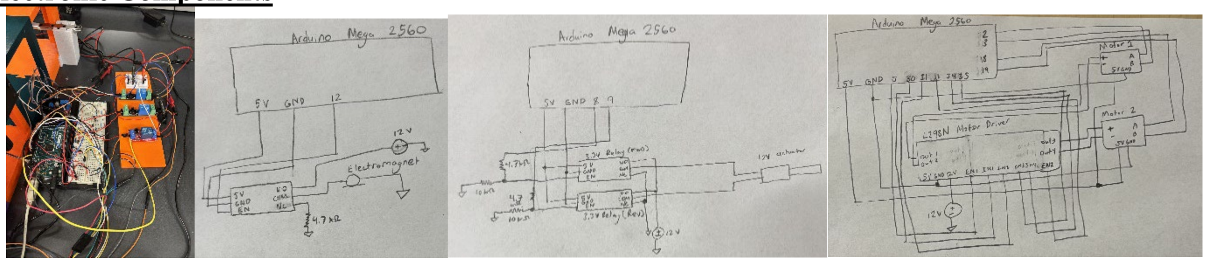

Figure 2: Physical wiring and subsystem wiring diagrams (Physical, Electromagnet, Actuator, DC Motors, left to right)

The gantry's motion is controlled by two DC motors, which are driven via an L298N motor driver. The driver receives a PWM signal from the Arduino to regulate the speed and direction of the motors. Each motor operates at 12V, with current draw depending on the load. Under typical conditions, the motor draws 1.8 A. The power consumed is calculated as:

$$P = V \cdot I = 12\,V \cdot 1.8\,A = 21.6\,W$$

The two voltage dividers reduce the 5V logic from the Arduino to 3.3 V, the relay switching voltage. The divider consists of two resistors, R_1=10 kΩ and R_2=4.7 kΩ, connected in series across the 5 V supply. The output voltage (V_{out}) is taken from the junction of the two resistors and is calculated using the voltage divider equation:

$$Vout \; = \; Vin \cdot \frac{R1}{R1 + R2} = 5 \cdot \frac{10k}{10k + 4.7k} = 5 \cdot 0.68 = 3.4V$$

To control the electromagnet, we used a 5V relay to switch the 12V electromagnet control voltage on and off. Our circuit diagrams for each subsystem are shown in Figure 2.

Mechanical Components

The actuator was responsible for dispensing game pieces onto the board. We selected a linear actuator with a maximum force rating of 120 N, as our load would require less than 120N. This selection ensured robust performance without overengineering the system. The actuator was ideal for the purpose of generating repeatable motion. The specifications of this actuator align with the required mechanical performance while maintaining cost-efficiency.

The electromagnet, selected for its strength and low power consumption, was powered at 12 V through an H-bridge motor driver. The L298N motor driver module was also chosen for its high current ratings and dual-channel operation. This module allowed control over the electromagnet's polarity, facilitating easy pickup and release of the game pieces. The Arduino Mega 2560 microcontroller, chosen for its high number of I/O pins and compatibility with various sensors and actuators, served as the central control system. It integrated seamlessly with 3.3 V and 5 V relay modules, DC motors, and a linear actuator, with the 3.3 V relay operating via a voltage divider to ensure proper voltage matching.

Our 20-tooth pulleys and 5mm timing belt provided consistent movement on our axes. We used linear slide rails for smooth and stable gantry movement. We chose 600 mm rails so our gantry could move across the entire board.

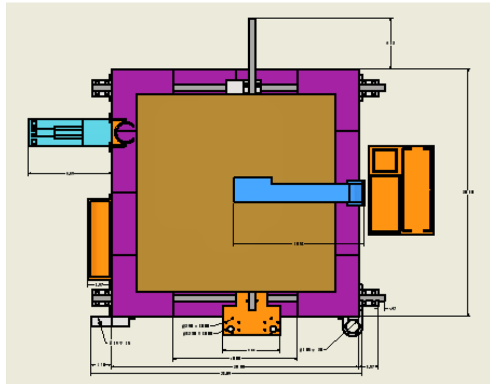

Figure 3: Dimensioned mechanical drawing

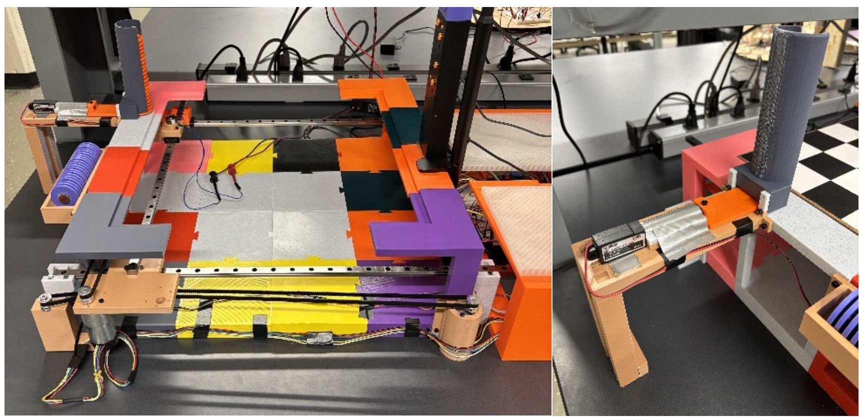

Figure 4: Pictures of the mechanical assembly

Our mechanical assembly consisted of the PLA base, PLA mounts, linear slide rails, DC motors, timing belts, and the linear actuator (Figures 3 & 4). Our total cost for these items was $227.83. To test our mechanical assembly, we sent the DC motors to various locations across the board. We made sure the gantry lined up with each expected board square after moving. We also made sure the DC motor PID gains were tuned since a load was added. Lastly, we made sure the actuator piece could dispense all the pieces in the tube with no issues.

Software Components

Figure 5: Simulink model for our final design

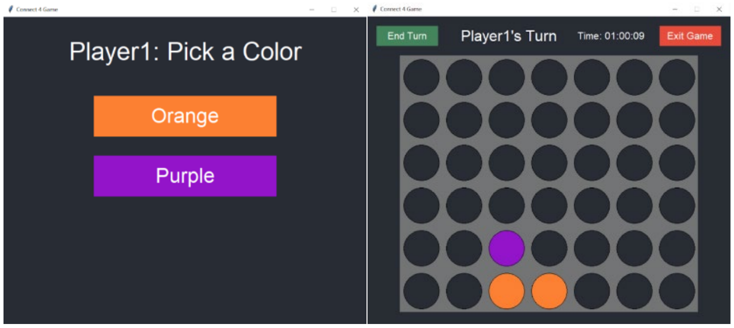

Figure 6: GUI design

The software design included MATLAB, Simulink, and Python. Python was used to develop the algorithm for an unbeatable robot and the Graphical User Interface (GUI), shown above in Figure 6, for the game. It communicated back and forth with MATLAB, which passed values to Simulink to interact with the hardware. In this software system, once the game is started, the user selects the color he or she wants using the GUI and if the user wants themselves or the robot to go first. The robot determines the best move depending on the location of the user's first move or the best starting move if the robot is meant to move first.

The Python code features a Minimax algorithm that determines the best move up to a depth of 4, meaning that it checks all possible moves and consequences of each move up to 4 moves in the future. It then sends the row and column to MATLAB, which controls the motors and moves them to the correct x and y positions (see Figure 5). The electromagnet is activated to drag the game piece to the position determined by the robot's algorithm and the selected best move. The algorithm then checks if either the robot or the player has won by looking for a sequence of four or more pieces in a row, either vertically, horizontally, or diagonally. If there is a winner, the game ends. If not, the robot waits for the player to make a move. Once the player makes a move and confirms it via the GUI, an image is taken to determine the location of the moved piece. The determined location is sent back to the Python file which allows the UI to update with the user's move. Finally, if the game is over with the same criteria mentioned earlier, then the game will end, but if not, the code will continue to cycle back to the robot, determining the best move and so on. Our software cost was $0 because we did not purchase any additional software.

<u>System Integration</u>

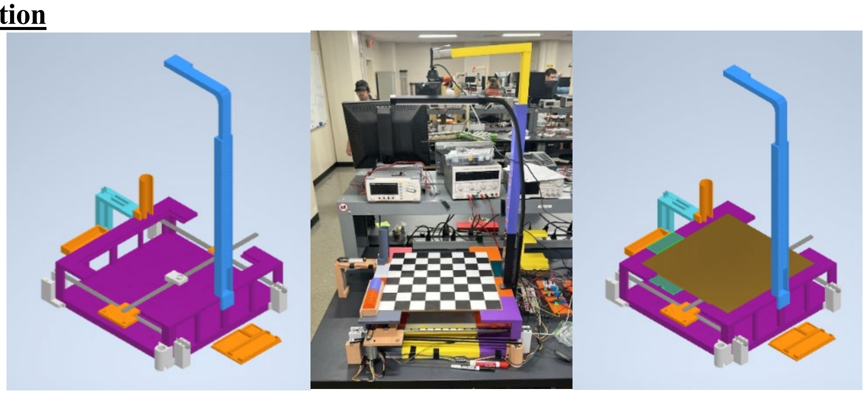
Figure 7: Assembly Diagram for the robot

To assemble our design, first screw in all the plastic components as shown on the left in Figure 7. Then, place the motors, belts, and other components as shown in the center picture. Screw those in as well and tighten the set screws on the pulleys. Put the electromagnet in the gantry. Attach the actuator, then wire the circuit as shown in Figure 8. Lastly, cover the inside by placing the game board on top.

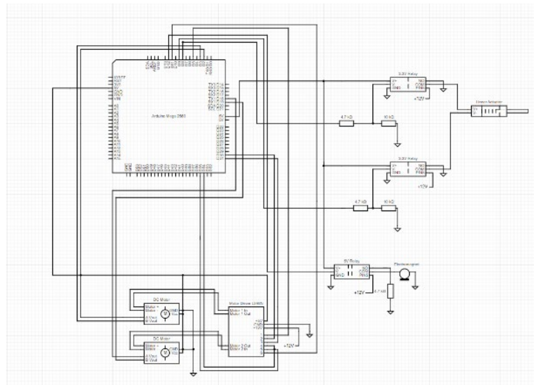

Figure 8: Connect 4 Robot Wiring Diagram

During testing, our system completed a full Connect Four game autonomously. However, we encountered issues during other test sessions. The motors would unexpectedly stop working and required a slight push to resume operation. Also, MATLAB 2022b crashed many times during test sessions. After evaluating these occurrences, we believe the root cause lies in our timing belt design. Our design inadvertently included a belt pinch point that created uneven tension. Our design's software interruptions revealed a need for better error handling in our motor movement code. Overall, while the system successfully played Connect Four autonomously, these challenges identified areas for improvement in both the mechanical and software design.

Our system costs $466.12, exceeding the initial budget of $300. Our failure to stay within the budget was partially caused by the required components, namely the Arduino Mega 2560. These components did help with reliability and precision, but we would replace them with more cost-effective components if we were able to choose. Additionally, sourcing components at per-item cost rather than bulk price, added to our expenses. Despite going over budget, these higher-quality components were essential for achieving our project goals.

Bill of Materials							
Item #	Item Name	Quantity Included in Purchase	Quantity Used in System	Item Link	Data Sheet Link	Total Cost	Per Unit Cost
1	Arduino Mega 2560	1	1	store-usa.arduino.cc	store-usa.arduino.cc	$ 48.40	$ 48.40
2	DFRobot DC Motor 251 RPM 12V	1	2	Digikey.com	DF ROBOT	$ 23.12	$ 46.24
3	Adafruit Industries Electromagnet 3873	1	1	Digikey.com	DF ROBOT	$ 9.95	$ 9.95
4	Logitech Brio 105 1080p	1	1	bhPhotoVideo	N/A	$ 39.99	$ 39.99
5	H-Bridge Motor Driver L298N	4	1	Amazon.com	Amazon.com	$ 9.99	$ 2.50
6	UpBright 5V AC/DC Adapter	1	1	Amazon.com	Amazon.com	$ 7.99	$ 7.99
7	Amazon Basics USB-A to Micro USB Charging Cable, 3 Foot	5	1	Amazon.com	N/A	$ 13.29	$ 2.66
9	Optocoupler 3/3.3 V Relay High Level Driver Module	5	2	Amazon.com	ibspot.com	$ 10.69	$ 4.28
10	Voncerus LED Desk Lamp w/ Clamp	1	1	Amazon.com	N/A	$ 18.28	$ 18.28
11	DC House Mini Electric Actuator	1	1	Amazon.com	Amazon.com	$ 32.09	$ 32.09
12	Neodymium Magnets (15 x 2 mm)	50	42	Amazon.com	N/A	$ 18.17	$ 15.26
14	WINSINN GT2 6mm Pulleys	5	5	Amazon.com	N/A	$ 6.41	$ 6.41
16	Metric Screw Assortment, Metric Machine Screws [M2 M3 M5]	1	1	Amazon.com	N/A	$ 23.53	$ 23.53
17	HOCPKOT 600 mm MGN12 Linear Rail Guide	3	3	Amazon.com	Amazon.com	$ 102.69	$ 102.69
18	HiLetgo 5V Channel Relays	2	2	Amazon.com	Amazon.com	$ 7.39	$ 7.39
19	HiLetgo 12V Channel Relays	1	1	Amazon.com		$ 5.89	$ 5.89
20	Total Filament Used for Housing and Additional Components	100	~40% in Final Design			$ 100.00	$ 40.00
21	Zeelo GT2 Timing Belt Pulley and Belts, 5mm, 20 Teeth, 2mm Pitch, 6 mm Wide	2	2	Amazon.com	Amazon.com	$ 32.08	$ 32.08
24	Eloooga Mulitcolored Dupont Wire 40 Pin Male to Male, Male to Female, Female to Female Jumper Wires	120	33	Amazon.com	Amazon.com	$ 5.99	$ 1.65
25	Morrison Games Color Gaming Tokens	50	24	Amazon.com	N/A	$ 9.53	$ 4.57
26	Chess and Checkers Board Sticker	1	1	Tenstickers.com	N/A	$ 14.23	$ 14.23
27	BOJACK 1000 Pcs 25 Values Resistor Kit 1 Ohm-1M Ohm with 5%	1000	5	Amazon.com	N/A	$ 9.99	$ 0.05
						Total Cost	$ 466.12

Table 1: Bill of Materials

Analysis of Final Prototype Performance

Our design succeeded in meeting most of the customer requirements. We dealt with challenges regarding 3D printing tolerances, and we were over budget by $166.12 (see Table 1).

The 3D printing tolerancing issues were due to the low precision 3D printers used at the campus library. Although they were cost-effective, we did not think the poor consistency was acceptable. If we were to create our robot again, we would make several changes to our design. We would fashion a base made from wood rather than multiple 3D printed pieces. The wood in the base would be more stable, and it would increase the quality of our design. It would also be more cost-effective. On the other hand, it would be heavier.

Our other issue was the result of the price and quantity of parts we purchased. We did not keep track of our budget during the design process, and we bought components one at a time. Since we checked our budget after everything was assembled, we could not go back to fixing the issue. If this were a mass-produced product, we would buy components in bulk, and the cost per unit would be lower. Also, we did not think about adding components such as the camera and Arduino to our cost analysis until the very end. They took a sizeable chunk out of our usable budget. If we were to design a new system, we would select components that only provide what we need instead of an all-around device like the Arduino Mega 2560.

Unfortunately, our timing belt design caused mechanical failures during robot operation. One of our DC motors acted as a moving belt tensioner. Once it reached the edges of our board, the belt was too tense in the smallest segment, and the motor could not provide enough torque to move the belt. Using a timing belt tensioner and clearing the belt's path would reduce the need to precisely tune our timing belts by bolting the belt to a new location each time.

Our printed component designs had many faults in the beginning. We had to re-design and reprint our components multiple times, changing infill levels, shapes, and sizes as necessary. We also had components that didn't work well together. They did not meet our tolerance levels. In future projects, we would plan to have a plan finalized prior to starting the project. This would allow our design to flow more seamlessly, have less time delays, and minimize the number of reprints.

Project Schedule/Gantt Chart

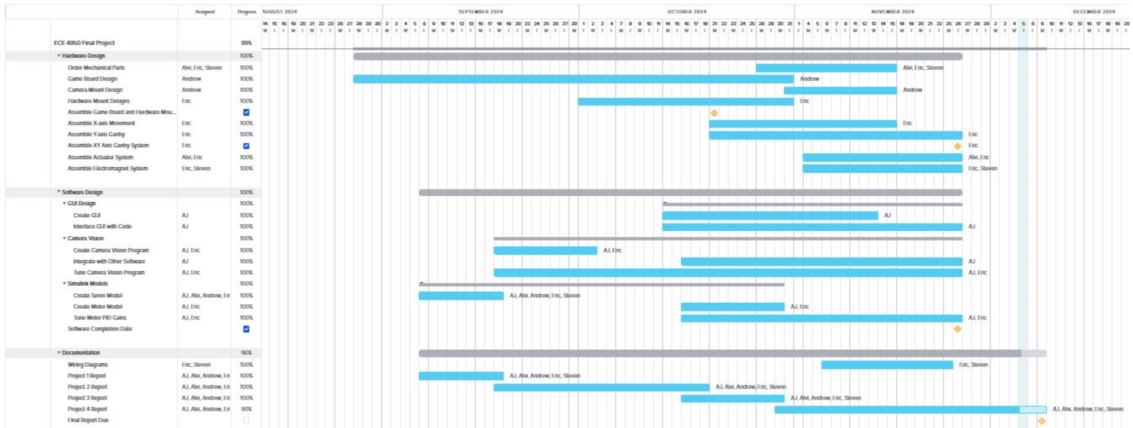

Figure 9: Project Schedule/Gantt Chart

Figure 9 shows our project's development over time.

ECE 4950 Project 4 – Customer Requirements and Final Design Parameters

Team 21

Group Member Last Names: Berkeland, Eubanks, Johnson, Mitchell, Topuzi

Score	Pts	
	5	**General Format - Professional Looking Document/Preparation (whole document)** a) Fonts, margins (11pt, times new roman, single spaced. 1" margins on all sides). b) Spelling and grammar are correct c) Layout of pictures – all figures need numbers and captions and must be referenced in the text d) Follows the page limitations below. e) References. Use IEEE reference format. f) This grading sheet is included as the final page.
	0	**Page 1: Title, Group Name, Group Members, and Date** **Executive Summary** (1 concise, well-written paragraph) Provide an overview of this project. Briefly describe what you did and what you learned.
	5	**Page 2: Engineering Requirements** (<1 page) Bulleted list of Final Design Engineering Requirements
	10	**Pages: 3-7: Design Details (<5 pages)** Describe a system that can be built including System Architecture and System Integration based on the Engineering Requirements. Do not include data sheets or software code.
	10	**Page 8: Analysis of Final Prototype Performance** (<1 page) Did it succeed or fail to meet customer requirements? What went wrong and what happened in the design process to allow this problem? Make a table of the customer requirements and address how well your design met these expectations.
	5	**Page 9: Project Schedule/Gantt Chart** (<1 page) Create a schedule (Gantt chart) that shows the tasks and schedule for your project. Start from the very beginning of your project and extend to the end (completing final report and presentation).
		Page 10 This grading sheet is included as the final page.
	50	Laboratory demonstration of your prototype (evaluated by instructor and TAs). Evaluator will manipulate the interface and evaluate how well the system provides the timing and display functions (i.e. how well does the closed loop control work). Is it well built? Neat wiring? (.6 * the prototype evaluation score)
	15	Rating by reviewers after competition
	15	Poster for Demo containing the following from the report: • Executive Summary • Customer and Engineering Requirements • Software Explanation • Prototype Design Explanation • Performance Analysis