CONNECT 4 ROBOT

A Research Paper submitted to the Department of Electrical and Computer Engineering
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Electrical and Computer Engineering

By

Jared Tyranski

December 17, 2021

Technical Project Team Members

Kellan Delaney
Roman Kaker

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISOR
Harry Powell, Department of Electrical and Computer Engineering

Third Times the Charge/ Connect 4 Robot

Kellan Delaney, Jared Tyranski, Roman Kaker

12/17/2021

Capstone Design ECE 4440 / ECE4991

Signatures

Statement of work:

Kellan Delaney

My primary role for this project was the firmware design for the MSP430 microcontroller. This included reading from the photo-interrupter to detect chip placement, writing PWM outputs to the servo motor and stepper motor controller, communicating with the game algorithm on the laptop via UART, and defining the overall flow of the game. I defined the custom instruction set that the laptop and microcontroller use to send information back and forth.

I was also responsible for 3D design and construction of the overall system. This included designing individual parts which were 3D printed, soldering together the photo-interrupters, and attaching everything together according to our design. The custom 3D printed parts that I designed are a funnel array that guide the chips into the board, the “sensor base” to which the photo-interrupters are attached, a chip tube which holds the robot’s chips, a gear and rack assembly that attaches to the servo and dispenses chips out of the chip tube, a mount for the PCB, and mounting arms which hold the Connect 4 board to the rest of the system. I also helped pick out parts for the printed circuit board (PCB).

Jared Tyranski

My primary role for this project was the design and construction of the hardware. This consisted of the schematics, design of the footprints, and routing the wires for the PCB. For this aspect of the project, my role also entailed to creation of the bill of materials (BOM) for the components, selecting the proper components and helping to order them, and even soldering a few pieces of the components onto the PCB if there were any manufacturing errors. For the schematic layout, I created custom symbols and footprints for components not already listed within Multisim. I made sure the board was easy to test either by including header pins, test point pins, and terminals with testable metals on them.

When the boards arrived, I took them to WWW Electronics Inc. (3W) to have all the components soldered. The boards were then tested with a digital multimeter (DMM) to ensure each hierarchical block of the circuit were verified and gave the results as expected. The power supply was also heavily tested to ensure that the PCB would get a proper amount of power for each subsystem in total.

Roman Kaker

My primary role for this project was the development of the game software that implemented the algorithm that the Connect4 Robot would use to determine its next move in response to the human players' movement. This involved implementing the minimax algorithm, a well-known game algorithm used in many games that implement a computer vs player feature. The development of this software entailed the use of many different API to take in input and produce output for the firmware on the MSP430 to provide instructions to the robot for chip placement. The code developed communicated via USB to UART cable that utilized a serial communication port to provide the code with input from the MSP430 which would then cause the software to produce a response to the human players move and output, through the serial communication port, the move that the computer generated through the use of the minimax algorithm. Additionally, I made it possible for the code to be executed as an executable, so any individual that had their hands on the robot could plug in the microcontroller into their laptop, run the executable Connect4 .jar file and play against the robot without any hassle. The executable is able to launch a command line window and run the game software to allow for gameplay to smoothly proceed without the need for explicit commands to be entered into the command line for the game to be played.

Table of Contents

Capstone Design ECE 4440 / ECE4991 1
Signatures 1
Statement of work: 2
Table of Contents 4
Table of Figures 6
Table of Tables 6
Abstract 7
Background 7
Constraints 8
Design Constraints 8
Economic and Cost Constraints 9
External Standards 9
Tools Employed 10
Ethical, Social, and Economic Concerns 11
Environmental Impact 11
Sustainability 11
Health and Safety 11
Manufacturability 12
Ethical Issues 12
Intellectual Property Issues 12
Detailed Technical Description of Project 15
Mechanical 16
Hardware 19
Firmware 24
Software 30
Project Time Line 32
Test Plan 34
Final Results 36
Costs 38

Future Work 39
References 40
Appendix A: Design Files and Code Links 46
Appendix B: Costs 47
Appendix C: Power Calculations 49

Table of Figures

Figure 1. System Block Diagram 15
Figure 2. System 3D Model 16
Figure 3. Chip Dispenser Model 17
Figure 4. Chip Detecting and Funnel Model 18
Figure 5. Mounting System Model 19
Figure 6. Main Schematic 19
Figure 7. Power Supply Subsystem 20
Figure 8. USB to UART Subsystem 20
Figure 9. Stepper Motor Driver Subsystem 21
Figure 10. Sensor Input Subsystem 21
Figure 11. PCB Layout Ground Plane Hidden 23
Figure 12. PCB Layout 23
Figure 13. Final PCB Populated 24
Figure 14. Firmware Overview 25
Figure 15. Servo Driver Functions 26
Figure 16. Stepper Driver Functions 28
Figure 17. Photo-Interrupter Functions 29
Figure 18. UART Functions 30
Figure 19. SerialPort Initialization 31
Figure 20. Command Line Interface 32
Figure 21. Original Project Timeline 33
Figure 22. Updated Timeline for Midterm Design Review 33
Figure 23. Final Timeline 34
Figure 24. Test Plan 34
Figure 25. Total Project Costs for Electronics 47
Figure 26. Total Project Costs for Mechanical Parts 48
Figure 27. BOM for One Connect 4 Robot 48

Table of Tables

Table 1. Pin Mappings 24
Table 2. Serial Instruction Set 29
Table 3. Grading Rubric 37
Table 4. Total Project Costs 38
Table 5. Costs to Produce One Connect 4 Robot 38

Abstract

Third Times the Charge's final capstone project is a semi-autonomous robot that is capable of playing Hasbro's Connect 4 [1] game against a human player. The system consists of a custom mechanical assembly that attaches to a Connect 4 board and can dispense chips into any column and an array of photo-interrupter sensors which detect when chips are placed into any column. This assembly is driven by a stepper motor and servo motor which are controlled by a microcontroller. A software program running on a laptop calculates the robot's next move and communicates with the microcontroller using a wired USB to UART interface. This project is an example of autonomous robots being used in consumer entertainment. Consumer robotics is a developing market that has seen increased growth during the years of the pandemic [2].

Background

The creation of technology that allows for automated responses to physical stimulus has grown rapidly, allowing individuals to interact with software and hardware as if they were interacting with another human being. The need for this type of interaction has increased, especially due to the pandemic, creating a sense of loneliness and other mental impairments [3]. This project addresses this issue by allowing isolated individuals to keep themselves mentally stimulated through gameplay with a robot. Additionally, the creation of such a machine is incredibly intriguing due to the many possibilities that can occur in a simple game of Connect 4. This project is a means of enabling society to enjoy their time when alone or socially isolated by playing against an algorithm in the physical world. It is one of many possible applications of robotics in everyday life.

One student at MIT completed a similar project [4]. The student also developed a robot that plays Connect 4 against a human, with configuration through a control panel of push buttons and switches. Contrary to that project, one goal of this Connect 4 robot was to be able to attach and detach a typical Connect 4 board made by Hasbro, such that a player could use the same board to play with the robot or another person. Another distinction is that this robot utilizes a computer program instead of a physical control panel to allow the player to choose who will go first and the difficulty of the robot, as well as display any game information, such as who won.

Many of the courses offered in the Electrical Engineering and Computer Engineering curriculum were needed to develop a working Connect 4 robot. From ECE Fundamentals I, II, and III (ECE 2630, 2660, 3750), knowledge of electronics, circuit design, and PCB design was used. Embedded software development techniques were used from Introduction to Embedded Computer Systems (ECE 3430), Embedded Computing and Robotics I and II (ECE 3501, 3502), and Advanced Embedded Computing Systems (ECE 4501). Hardware logic knowledge was used from Digital Logic Design (ECE 2330). Software development techniques and data structure knowledge were used from Software Development Methods (CS 2110), Program and Data Representation (CS 2150), and Advanced Software Development Techniques (CS 3240). Probability knowledge was used from Probability and Stochastic Processes (ECE 6711). All three members of Third Times the Charge (Kellan Delaney, Jared Tyranski, and Roman Kaker), are enrolled as double majors in Electrical and Computer Engineering, allowing us to have knowledge of the fundamentals of electrical engineering as well as the core concepts needed to develop code as both embedded firmware and pure software. As for specific experiences, Roman is more geared towards developing software and code for embedded systems due to his experience as a Software Engineer intern. Jared, on the other hand, is more in tune with hardware design due to his time spent as an intern for hardware design and PCB design for embedded systems combining mechanical machines with electrical engineering. Kellan has experience building hardware and software systems around microcontrollers in various applications from his internship experience and extracurriculars, as well as hobby level 3D design experience. Although each of us has our own specialties, all members of the team had knowledge of every section of the project to ensure a dynamic approach could be taken to any impediments the team faced.

Constraints

Design Constraints

Several constraints were put in place for the design by the capstone course. The project must contain a microcontroller and include a custom printed circuit board designed by the team. In addition, there were three dates when a PCB design could be sent out to be printed: October 8, October 22, and November 12. Out of those three, the team could utilize two send out dates. This meant that the final PCB design had to be completed by November 12.

CPU Limitations

The Texas Instruments MSP430FR2433 [5] microcontroller was selected as the CPU because it had the necessary peripherals that were needed for the project, while also not being excessive for the application. The necessary features that were needed were 10 GPIO pins (7 photo-interrupters, 1 limit switch, 1 stepper direction, 1 stepper enable), 2 PWM outputs (servo and stepper), and 1 UART module. 16 out of 20 pins were used on the LaunchPad Development Kit for this microcontroller [6] in this project.

Software Availability

National Instruments' Multisim [7] and Ultiboard [8] were used for circuit schematic and PCB design because UVA had active licenses for them. Code Composer Studio [9] was used to develop embedded firmware because it is free and is compatible with the selected microcontroller. Visual Studio Code [10] was used to develop the laptop program because it is free and is easily customizable to work in a variety of programming languages. Autodesk Tinkercad [11] was used to create 3D designs because it is free and very simple to use. MakerBot Print [12] was used to slice the 3D designs for printing because it is free and compatible with the MakerBot Replicator+, which was the 3D printer used for producing the custom parts.

Manufacturing Limitations

Manufacturing constraints were imposed by the PCB manufacturer, Advanced Circuits, to qualify for the two-layer special pricing of $33 [13]. The most relevant constraints to this project were:

- Minimum 5 mil line/space
- Minimum 10 mil hole size

Parts Availability

The availability of our selected parts was dependent on the stock of electronics supplier DigiKey Electronics [14] for our electronic components and OpenBuilds Part Store [15] for our linear actuation structure and support rails. The only part that proved to have limited stock and was difficult to acquire were the photo-interrupters. Parts were chosen based on pricing and power consumption based on the chosen power source of a wall outlet.

Economic and Cost Constraints

The budget for this project was $500. Many of the more expensive parts, such as the stepper motor and servo motor, were sourced for free from the supply in UVA's NI Lab and team members' personal collections. Given that, running out of money was not a major concern, but the budget was still closely tracked.

If this device were to go into production to be sold, the cost may fluctuate depending on the cost of each of the individual parts supplied by manufacturers. Given that this product would likely cost significantly more than a typical board game, it would be impractical to market it towards individual families or consumers. Instead, this product would likely be targeted towards schools where a single device could be exposed to many children so that they can have exposure to robotics and to have fun.

External Standards

According to the Consumer Product Safety Commission of the US government, products that have a diameter smaller than 31.7 mm and fit within a cylinder of varying height from 25.4 mm to 57.1 mm are considered a choking hazard [16]. While a connect 4 chip is about 33 mm, this would still be a product that should not be played with kids under the age of three since it could still be possible for them to choke on such an item.

Since the program will be using C for the code of the algorithm and embedded programming, this will follow the Barr C coding standards to account for less bugs and for formal standards for ease of coding and communications [17].

Another standard will be for the type of circuit. As a class 1 circuit, the standard is that the output power of the circuit does not exceed 30V or 1000 volt-amps, and it can use a 120V power source from an outlet as preferred by OSHA standards [18].

PCB designs and standards are set by the IPC where the part and track spacing was considered and set to at least the appropriate minimum distance spacing based on the IPC-2221A

[19]. For materials for multilayer boards, the used materials were made accustomed to the standards of IPC-4101C for the base material such as laminate [20]. Finally, the criteria for the PCBs such as the material, solder mask, holes, board edges, and plating were designed to fit the standards for IPC-A-600J [21].

The EU standard for electrical and electronic products comes from the Restriction of Hazardous Substances (RoHS), and in this case most of the components used were RoHS compliant which is useful for reducing the environmental consequences of the design and production of the PCB with all the components used [22].

The Universal Asynchronous Receiver-Transmitter (UART) is a circuitry for serial communication between a transmitter sending the data and a receiver taking in the data [23]. Since there are some standard baud rates necessary for the use of UART, consideration of the baud rate led to the choice of 115200 bits per second to send data from the laptop to the PCB and ultimately to the MSP.

The USB standards are set by the USB Implementers Forum for the communication between a peripheral device and a computer [24]. The USB was used for communications between the laptop and the PCB when transferring the code, so the guidelines were followed to ensure that the communication had no issues occurring from a standards perspective.

Tools Employed

Several programming, design, and testing tools were used to complete this project. The tools for each distinct section of the project are described in detail below.

Mechanical

To design and model the system and individual parts in 3D space, Autodesk Tinkercad [11] was used. MakerBot Print [12] was used to slice the 3D models in preparation for 3D printing, which was done on the MakerBot Replicator+ [25] accessed through UVA's 3D-Printing Studio [26]. Kellan had previous experience with 3D modeling and printing before this project, but this was his first time using this specific toolchain so there was a small learning curve.

Hardware

To design the circuits and layout the printed circuit board, National Instruments' Multisim [7] and Ultiboard [8] were used. Multisim was vital for designing the schematics for the circuit board allowing for a utility towards component design and structuring. The skills Jared improved on were the extra utilization of the hierarchical blocks to structure each subsystem within the circuit board. The skills Jared improved on were the use of the component wizard since a lot of unique components were used for this project, and so the component wizard allowed for a way to create these components on Multisim to be able to lay out the circuit. For Ultiboard, this application allowed for a way to design the layout of the PCB so that it could be printed and used for the actual system of the Connect 4 Robot. The skills Jared had to improve on this application were the utilization of the part wizard which allowed for the creation of new footprints of unique components. With this skill learned, any new design could be implemented on the board with the correct drill holes and dimensions to ensure that the board could maintain each component and fit them when soldered onto the board.

Firmware

The firmware was written in C using Texas Instruments' integrated development environment (IDE), Code Composer Studio [9]. The firmware utilized the MSP430 driver library for the MSP430FR2XX_4XX family [27] as well as *stdint* [28]. Kellan has worked with CCS before with other Texas Instruments microcontrollers, so this was a familiar toolset for him.

Software

The game algorithm and communication software were written in Java using Microsoft's Visual Studio Code [10]. The software utilized the *java.util* [29] and *java.io* [30] packages as well as the *jSerialComm* library [31] for communication over a COM port.

Ethical, Social, and Economic Concerns

Environmental Impact

The main environmental impacts come when the device is thrown away and when the components of this device are manufactured. All the electronic components are RoHS compliant, which means their production minimizes use of hazardous products [22]. The aluminum rails and corners and steel screws are all recyclable and reusable [32]. It is unlikely that the electronics or parts in them can be easily recycled or reused by a consumer, but they can take them to an electronics recycler for them to decide. The PLA plastic parts from 3D printing are unfortunately not recyclable. Besides that, the game can be replayed and use the same pieces so there are not parts that need to be disposed of with each replay.

Sustainability

With the use of an outlet for an electrical source along with simple motors and mechanisms will allow for this device to have a long lifespan of playtime before the user would ever need to buy a replacement of the device due to electrical issues. Mechanical hardware should not deteriorate any time before there are electrical problems. The only constraint with sustainability is for the motor on the railing to function without any disjointed movement or staggering. If mechanical parts break down over time, replacements could be made to repair the device in the future making it mechanically repairable. For electrical parts, sensors could be replaced if any break, and the PCB could be swapped out if there is electrical or physical damage to the board.

Health and Safety

Since the device is meant to be a board game, the main safety concern is regarding protections for children. The tokens pose a hazard to kids younger than 3 because this would be a choking hazard, so this device should not be around infants and very young children [16]. The assembly of placing the device onto the Connect 4 board should be done by an adult since a child could potentially injure themselves if they struggle to assemble the game together. There are exposed motors and electronics, so it is advised that if children use this device with adult supervision. To protect users and the device from electrical shock, insulated wires are used whenever possible, and heat-shrink tubing is used to cover up any other exposed wire. The most dangerous wires are connected to the stepper with 12 volts at approximately 1.6 amps and the servo with 5V at approximately 1 amp. These levels are not particularly dangerous, but the wires are still completely insulated to protect the user.

Manufacturability

The design has many custom parts so it would be difficult to manufacture on a large scale without optimization in the design. All the 3D printed parts and the PCB are custom, so those would have to be specially made for each device. Everything else is directly available from a manufacturer, but they are still vulnerable to fluctuations in price and availability.

The construction of each device is also a moderately difficult task that would require a skilled technician to perform. All the photo-interrupters are soldered together for power. Many of the pieces are attached to each other using simple adhesives like hot glue, epoxy, and Command Strips [33]. The mechanical support structure is also hand screwed together, which might be difficult to automate given some odd angles that are necessary to get the proper force on the screws. The design for this robot would need to go through some serious rework if it were to go into mass production.

Ethical Issues

Since this device will be focused on playing a game against a human opponent without the use of data collection or profiling, there should be no ethical issues regarding race, gender, ideology and so forth. The only issue may be economic if this were to be produced since this would be an expensive add on to the original game, and so areas where many people who have incomes that are below the federal poverty line may not have access to this game.

Intellectual Property Issues

[34] presents a patent for a "Robot computer chess game". The technology presented is classified as an "Electric board game". The field of invention of the patent is "directed to a computer chess game, and more particularly, to a robot computer chess game capable of simulating humanoid characteristics while playing an expert level of chess." The patent's main independent claim is "In a computer controlled chess game having a housing member, a chess board, chess pieces, a computer processing circuit, a memory unit, a robot arm movably mounted adjacent the chess board, and means on the robot arm for grasping and releasing a chess piece, the improvement comprising: a multiple layer switch member positioned beneath each chess square to indicate the presence or absence of a chess piece; a magnetic member for each chess piece; a plurality of cavities beneath said multiple layer switch member and the chess board corresponding to each chess square and having a magnetic member movably contained therein, the cross-sectional area of said cavity configured to at least conform to the cross-sectional area of said magnetic member of the chess piece wherein placement of said chess piece actuates said switch member and spatially positions said chess piece at a predetermined location on each chess square to ensure alignment for coaction with the grasping and releasing means." This is backed up by a dependent claim that "the robot arm is articulated into at least two portions and further including means for coordinating the relative movement of each articulated portion of the robot

arm to ensure that the relative movement starts and stops at approximately the same time." By analyzing these claims and reviewing the capstone discussed, the robotic Connect4 game is still patentable because the system implemented by Third Times the Charge utilizes a railing system, instead of a robotic arm to place chips into the respective positions determined by the game algorithm. The other independent claims are related to the development of the robotic arm that is implemented in the patent and are unrelated to the capstone at hand.

[35] presents a patent for an "Intelligent game system for putting intelligence into board and tabletop games including miniatures". The field of invention of the patent is related to "the field of board and tabletop games including miniatures. More specifically, the present invention relates to the intelligent game systems for putting intelligence into board and tabletop games including miniatures." The patent's main independent claim is "A game piece for use in a board game having a set of rules and including a game board with a plurality of sensors, the game piece comprising: a figurine having a body including a base and an exterior shell coupled to the base, wherein the body is physically coupled to an opto-detector that is housed by the exterior shell, the exterior shell having a shape and color of one of the characters of the board game; object information stored on a memory and configured to be read by one of the sensors of the game board of the board game, wherein the object information includes an identifier; a rotation mechanism coupled between the base and the exterior shell and configured to rotate the exterior shell with respect to the base; and a light emitting source housed within the exterior shell and configured to selectively output light having a wavelength that is invisible to the human eye; wherein the rotation mechanism rotates the exterior shell with respect to the base based on a location of the game piece on the game board, an identity of the game piece indicated by the object information and the set of rules of the board game." This is backed by a dependent claim that the device will be comprised of "light transmission equipment". By analyzing these claims and revieing the capstone discussed, the robotic Connect4 game is still patentable because the system implemented by Third Times the Charge uses unique identifiers for each of the players pieces, however the identifiers are simply just black and red chips instead of each piece being unique. The other independent and dependent claims are related to the construction of the mechanical system that will move around the unique identifying pieces to progress the game forward.

[36] presents a patent for a "Coin depositing and dispensing machine". The field of invention "relates to a coin depositing and dispensing machine enabling depositing and dispensing of coins." The patent's main independent claim is "A coin depositing and dispensing achine comprising: a machine body having a front face, a rear face, and a first face and a second face in lateral direction of the machine body; a coin receiving port for receiving coins from the outside of a machine body of the coin depositing and dispensing machine; a coin ejecting port for ejecting coins outward from the inside of the machine body; a feeding unit which includes a rotary disc rotated around a rotary axis attached to the machine body at a position tilted at a predetermined angle in relation to a horizontal direction and a hopper for accommodating coins, which are received from the coin receiving port, on a surface side of the rotary disc with the coins not aligned, and can feed the coins, which are received from the coin receiving port, one by one; a depositing and transporting unit which includes depositing and transporting pulleys each

rotated around the rotary axis for pulleys attached to the machine body, at a position tilted at a predetermined angle in relation to the horizontal direction, and an endless depositing and transporting body supported by the depositing and transporting pulleys so as to extend along a first face of the machine body and having projections capable of pushing and transporting coins one by one, and transports the coins fed from the feeding unit; a recognition unit for at least recognizing denominations of coins transported by the depositing and transporting unit; a plurality of coin accommodating and ejecting portions which are successively disposed from the front side to the rear side of the machine body, includes accommodating and ejecting belts which are rotated from the first face to the second face of the machine body, and can accommodate coins, which are sorted for each denomination in accordance with a recognition result by the recognition unit on the accommodating and ejecting belts, for each denomination and eject the coins to the second face of the machine body when the accommodating and ejecting belts are rotated; sorting units which, by electric driving, in accordance with the recognition result by the recognition unit, discharge coins, which are transported by the depositing and transporting unit and cannot be accepted in the machine body, outward from the machine body, and sort coins, which are transported by the depositing and transporting unit and can be accepted in the machine body into the plurality of coin accommodating and ejecting portions for each denomination; and a dispensing and transporting unit which faces the depositing and transporting unit, extends along the second face of the machine body, and transports coins ejected from the coin accommodating and ejecting portions to the coin ejecting port." This is backed by a dependent claim that "The coin depositing and dispensing machine according to claim 1, wherein the dispensing and transporting unit includes: dispensing and transporting pulleys pivotally supported in a horizontal direction in relation to the machine body and rotatable; an endless dispensing and transporting body which is supported by the dispensing and transporting pulleys and can receive and transport a plurality of coins; and dispensing passage walls surrounding a coin transporting face of the dispensing and transporting body from its both sides and above. By analyzing these claims and reviewing the capstone discussed, the robotic Connect4 game is still patentable because the system implemented utilizes a servo along with a rotating gear and pusher to dispense a single chip at a time instead of a pully system that is capable of dispensing multiple coins at a time instead of a single coin.

Detailed Technical Description of Project

The goal of the project was to design and build a semi-autonomous robot that is capable of playing Connect 4 against a human. The system has been broken down into the following sections:

1. Mechanical
    a. Chip Dispenser
    b. Chip Funnel and Detecting
    c. Mounting
2. Hardware
    a. Components
    b. Schematics
    c. PCB Design
3. Firmware
    a. Servo Output
    b. Stepper Output
    c. Photo-Interrupter Input
    d. Communication
4. Software
    a. Game Algorithm
    b. Communication
    c. Interface

A high-level system block diagram is shown in Figure 1.

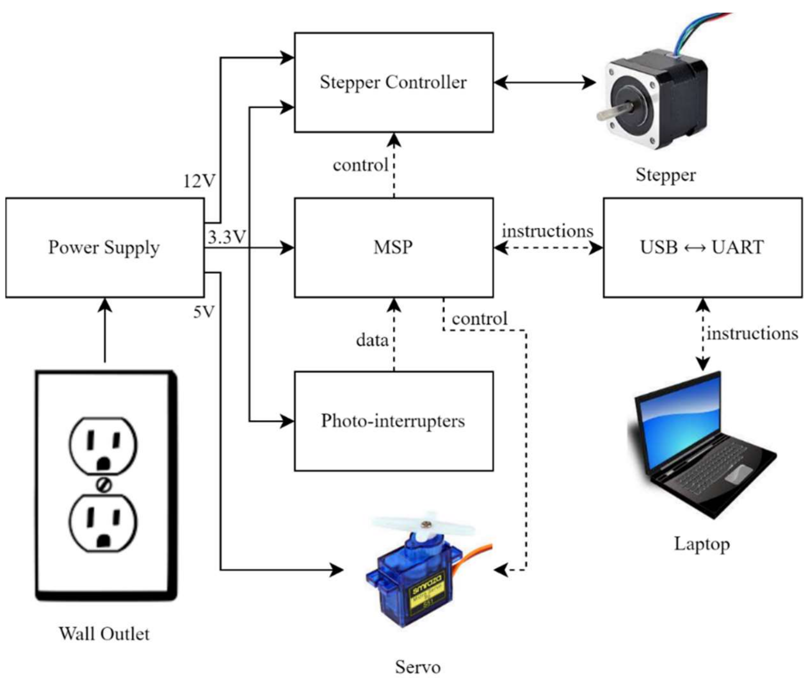

Figure 1. System Block Diagram

Mechanical

The mechanical design of the system falls into three general categories: the chip dispenser, the chip detecting and funnel system, and mounting. All the 3D modeling for custom parts was done using Autodesk Tinkercad, and all the 3D printing was done on a MakerBot Replicator+. A 3D model of the entire system is shown in Figure 2.

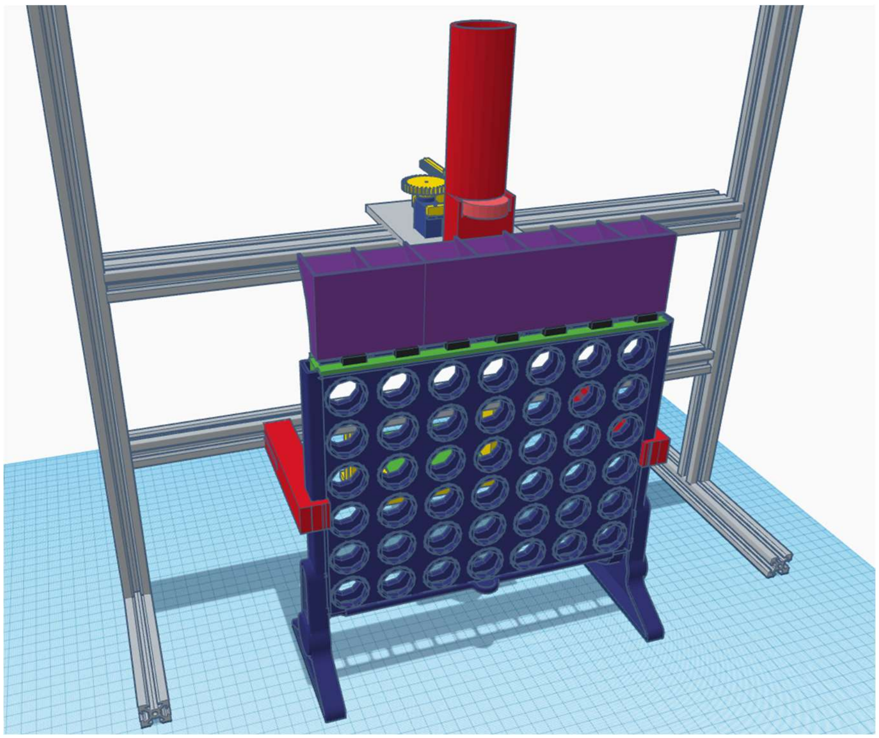

Figure 2. System 3D Model

Chip Dispenser

The chip dispenser is primarily made up of a cart on a linear rail that is belt-driven by an OpenBuilds NEMA 17 stepper motor [37] and an Smraza micro servo motor [38] on top of that cart. The stepper and linear rail allow the entire dispensing assembly to move horizontally across the top of the Connect 4 board such that a chip can be placed into any column. Attached to the servo is a 3D printed gear and rack assembly which translates the rotational motion of the servo into linear motion in the rack. A 3D printed chip tube is placed in front of the servo assembly which holds all the chips that the robot will use to play the game in a vertical stack. On one side of the chip tube is a small opening which allows the rack from the servo assembly to push a single chip out of a larger opening on the other side of the tube. The rack also holds up the rest of the chips in the stack as it pushes out the one on the bottom such that only one chip falls out of the tube at a time. The gear rack mount is attached to the servo using M2 screws and nuts. The gear is attached to the rotating shaft of the servo using Loctite two-part epoxy [39]. The chip tube and servo assembly are attached to the cart on the linear rail using Command Strips. An image of the chip dispenser model is shown in Figure 3 with the stepper and linear rail shown in gray, the gear and rack assembly shown in yellow, and the chip tube shown in red.

The servo assembly and chip tube had to be redesigned multiple times in order to successfully dispense the chips and have them dispense consistently. This current system seems to be very consistent and reliable.

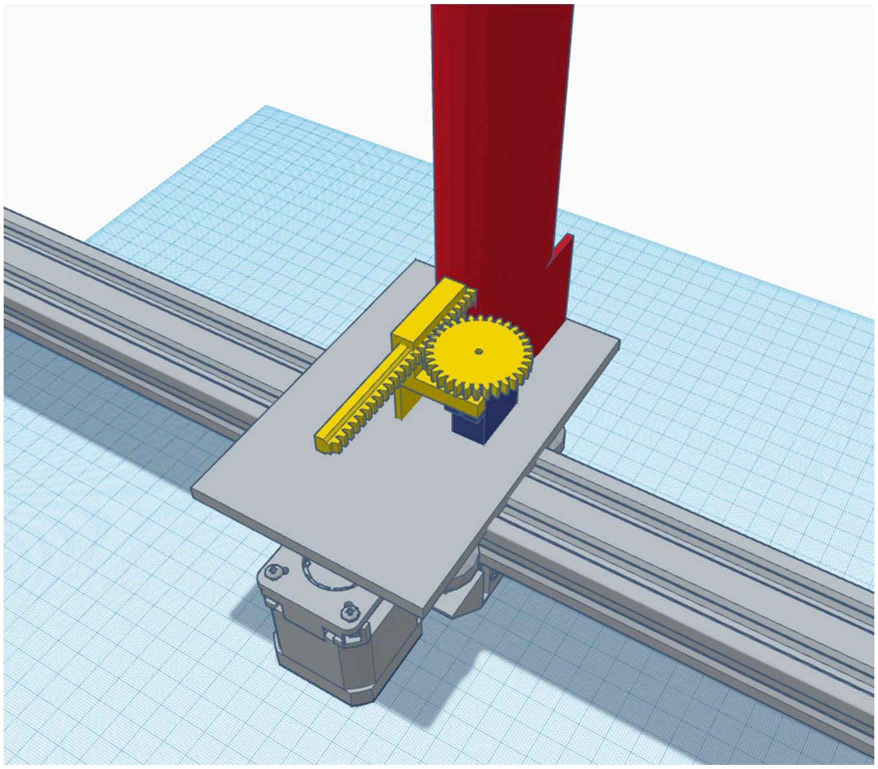

Figure 3. Chip Dispenser Model

Chip Detecting and Funnel

The chip funnel and detecting system is placed directly on top of the Connect 4 board. On the bottom layer, touching the board is the 3D printed sensor base. The sensor base acts as mount for the GP1A57HRJ00F photo-interrupters [40] such that they do not have to be directly attached to the Connect 4 board and are still removable if desired. The sensor base stays secure on top of the Connect 4 board by putting legs on either side into holes at the top of the board. On top of the sensor base, the photo-interrupters are aligned such that there is one on top of each column, and their beams cross approximately the center of each column in order to detect when the chips have fallen through. On top of photo-interrupters is the 3D printed funnel array, which act as a receptacle for the falling chips when they are ejected from the chip dispenser. The funnel array catches the chip, adjusts it from horizontal to vertical, and aligns it with the column in the Connect 4 board, allowing it to fall between the photo-interrupters, through the sensor base, and into the board. The sensor base, photo-interrupters, and funnel array are attached to each other using hot glue. An image of the chip detecting and funnel model is shown in Figure 4 with the sensor base shown in green, the photo-interrupters shown in black, and the funnel array shown in purple.

Constructing this system was a major challenge. First, the photo-interrupters had to be soldered together so that they only needed one power source each for all of the detectors and emitters. Then, they had to be hot glued to the sensor base, while taking special care to ensure that they do not block the gaps that the chips fall through. Next, the small pieces of plastic that attach the detector and emitter sides of the photo-interrupters had to be removed because they blocked the gaps. Finally, the funnel array was hot glued on top of the photo-interrupters, making sure to align it properly. This process had lots of opportunity for things to go wrong. There could have been an alignment issue, or the photo-interrupters could have been broken in the process. Luckily, everything was attached correctly on the first try, meaning the chips could fall through everything seamlessly and the photo-interrupters still worked.

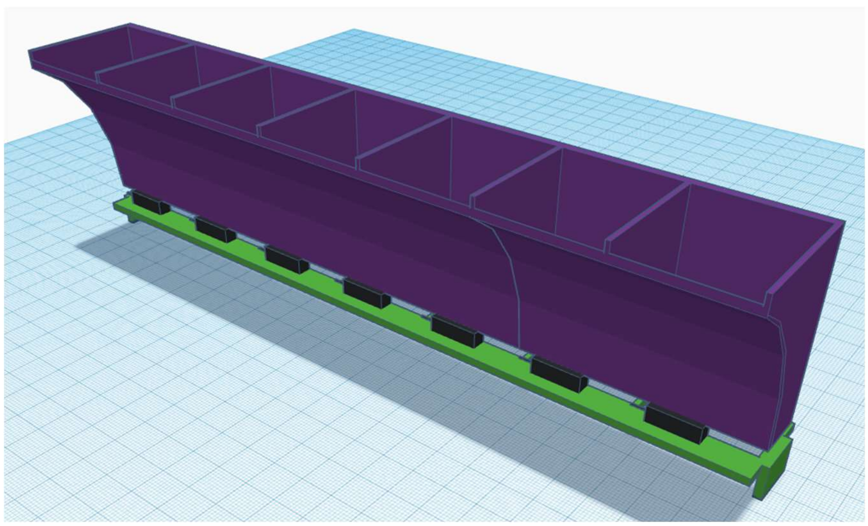

Figure 4. Chip Detecting and Funnel Model

Mounting

The mounting system is responsible for holding up the chip dispenser, holding the Connect 4 board in place, and holding the MSP Launchpad and PCB in place. OpenBuilds V-Slot Linear Rails [41] are used to support the chip dispenser. Attached to those linear rails are two mounting arms which attach to the side of the Connect 4 board and keep it in place. Also attached to the linear rails is also the PCB mount. The linear rails, mounting arms, and PCB mount are attached to each other using corner brackets, and M5 screws and nuts. The PCB mount attached to the MSP Launchpad with Command Strips. The mounting arms grab on to the Connect 4 board using a snap-fit and are designed to be attached and reattached with some force. An image of the mounting system model is shown in Figure 5 with the linear rails shown in gray, the mounting arms shown in red, and the PCB mount shown in yellow.

Making the mounting arms able to be reattached was a minor challenge that took some prototyping to ensure the fit on the Connect 4 board was correct.

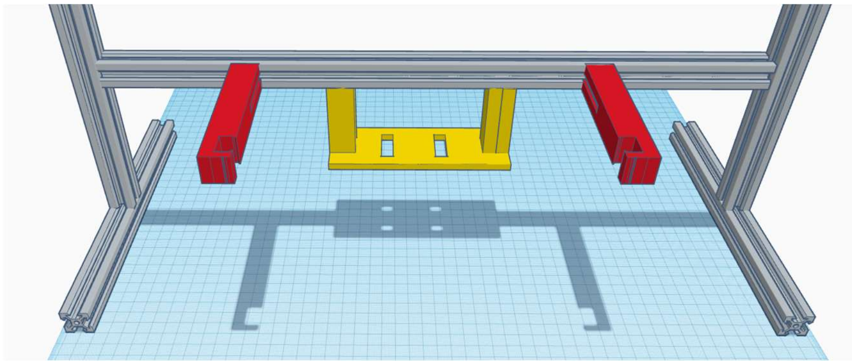

Figure 5. Mounting System Model

Hardware

The hardware consists of a PCB which contains a power supply which powers all the other subsystems including the stepper motor driver, the sensor input, the USB to UART converter, and connections to serve as a booster pack to the MSP microcontroller. The development of these subsystems is described in the following sections.

Schematic

The schematics for the board were developed in Multisim and were designed as a booster pack for the MSP430FR2433 [5]. This one was chosen because it had the necessary features that were needed such as 10 GPIO pins (7 photo-interrupters, 1 limit switch, 1 stepper direction, 1 stepper enable), 2 PWM outputs (servo and stepper), and 1 UART module. The total pins for this microcontroller were 20, and 16 pins were used in total. The main schematic consists of the booster pack for the MSP which is essentially two 10 pin receptacles [42]. The main schematic also has six white test points [43] to test various features of the MSP and PCB such as the Bump Switch, the MSP transmitter and receiver, and the PWM, Direction, and Fault of the stepper motor driver. There are also four black test points [44] as a ground reference for testing. There is a 3-pin male header [45] which acts as the connection between the MSP and the servo motor, and a 4-block terminal [46] which acts as the connection between the stepper motor driver and the stepper motor itself. Figure 6 displays the main schematic as detailed with all the components mentioned colored green with the hierarchical blocks shown.

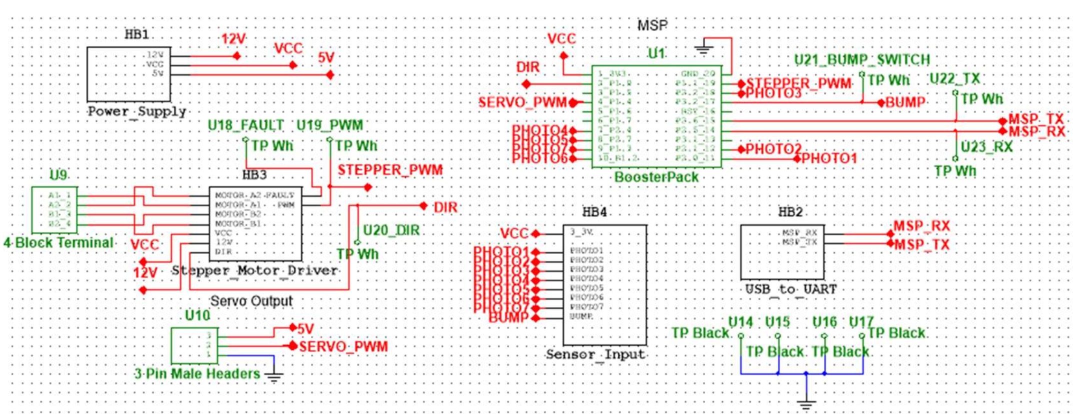

Figure 6. Main Schematic

The power supply is meant to supply different voltage levels of power to different systems and subsystems which was done by utilizing a 12V 54W wall outlet transformer [47] and connecting it to a power jack connector [48]. The power was then distributed at 12V, 5V, and 3.3V using voltage regulators such as the 5V linear regulator [49] and the 3.3V linear regulator [50]. The 3.3V regulator utilizes a 10µF bypass capacitor and a 100µF capacitor for power line stability while the 5V regulator utilizes a 0.33µF bypass capacitor and a 0.1µF capacitor for power line stability. The red test points [51] included here are used to test if the voltages supplied from each source gives the required voltage as needed for the system. Figure 7 displays the power supply schematic.

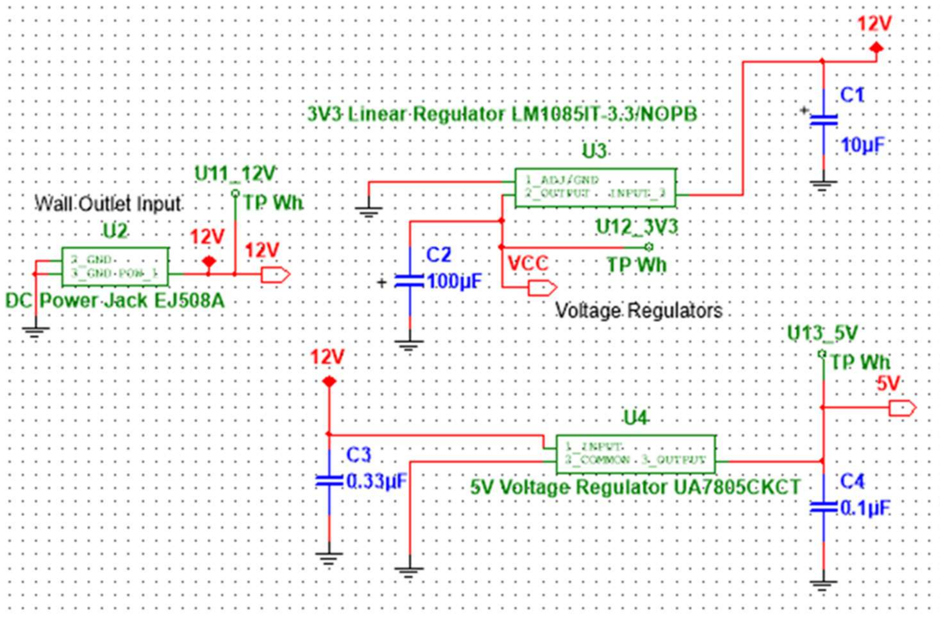

Figure 7. Power Supply Subsystem

The USB to UART subsystem is shown below in Figure 8 where the connection between the MSP receiver and transmitter is managed between the 6-pin male header [52] where a USB-UART cable [53] connects to the pins and the laptop. Originally the design attempted to create a USB to UART converter utilizing a chip and components, but this did not gather the required results, and so the cable connection was chosen as a simplified way to ensure that the USB and UART connections could be directly connected between the laptop and the MSP as much as possible.

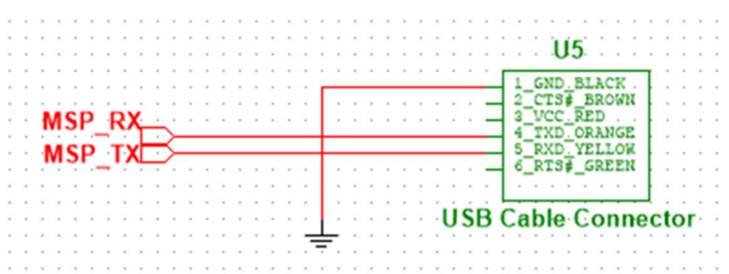

Figure 8. USB to UART Subsystem

The Stepper Motor Driver subsystem is shown below in Figure 9 where there is a 100µF bypass capacitor as recommended by the data sheet for the Pololu Stepper Motor Driver Carrier board [54]. There were two 8-pin receptacles [55] such that the carrier board could easily be placed and removed from the circuit board as needed for testing. The carrier board was chosen as a simplified way to utilize the stepper driver motor since originally the first design attempted to create a stepper driver motor with all the components. This first design did not gather the results required, and so this was chosen to simplify the design since it had the same current and voltage requirements for the stepper. The trimmer potentiometer on the carrier board is set such that the output current to the motor coils is approximately 1.5A, which remains under the motor's maximum of 1.6A, and is the maximum the carrier board can output without requiring a heatsink.

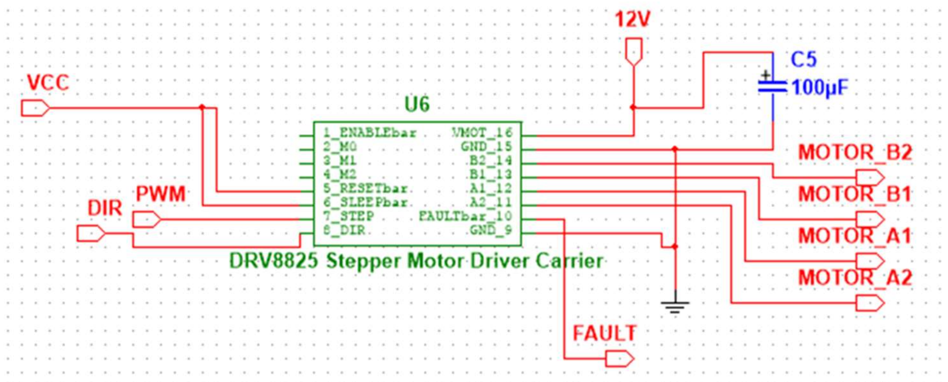

Figure 9. Stepper Motor Driver Subsystem

The Sensor Input subsystem is shown below in Figure 10 where there is a 3-terminal block [56] for a connection to the limit switch (bump sensor) [57] and the MSP. There is also an 11-block terminal [58] to connect the photo-interrupters [59] to the MSP.

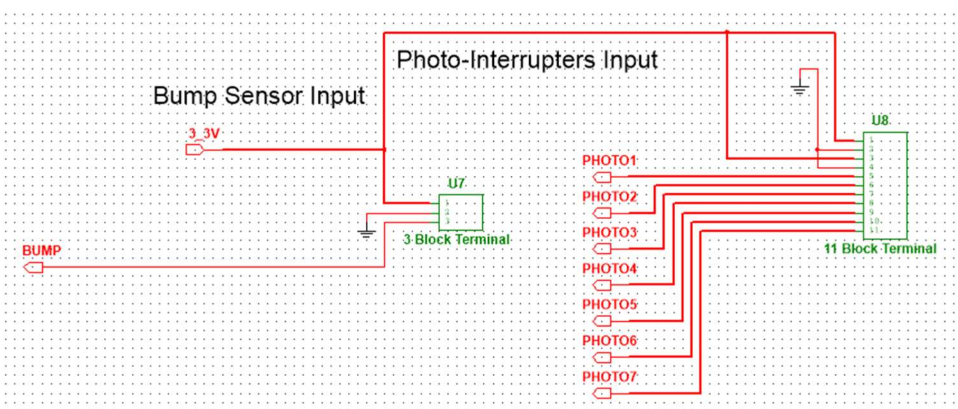

Figure 10. Sensor Input Subsystem

PCB Design

The PCB layout design was done in Ultiboard. This board was created on a two-layer board to maintain simplicity and ease of production since the Connect 4 Robot is supposed to mimic a mass production of a board game toy. The top layer contains the ground plane and traces that could be made on the top layer to maintain efficiency of tracing. The bottom layer contains traces that were unable to be routed on the top layer due to the ground plane already being a feature of that layer.

The layout of the components for the whole system was based on keeping the subsystems closely connected while maintaining ease of testability and construction of the final product. The connections for the MSP were kept in the center for stability purposed of the board while the terminal blocks were kept at the edges so that the wires could easily connect to the blocks since they faced the outside of the board. The power jack connector was also kept to the side for ease of connecting the power source into the PCB. The voltage test points were kept close to the power source and voltage regulators as an easy identifier of what those test points were while the ground test points were spread to the corners of the PCB for accessibility of the ground test point from any part of the PCB. The white test points were kept close to the reference location to keep traces for them to a minimum distance.

One problem that the design did not account for was the voltage regulator placement since these designs have a large ground metal piece on the back of the component to place onto to board for safety purposes to ensure as much surface area as possible for the ground of the voltage regulators. While the system is still functional and able to work even without these ground metal pieces connected to the PCB, it is an important lesson for future production of these PCBs with the components to correct this error.

The header pins were kept close to relevant locations in terms of proximity such as the 3-pin male header that was kept close to the 5V regulator since that was its source of power for the servo motor. The 6-pin male header was kept close to the MSP receiver and transmitter since this component was meant for the USB to UART connection. The capacitors were kept close to the devices they were paired with such as the voltage regulators and the stepper motor driver since it is important for the capacitors to remain close to the relevant components.

A few vias were used within the design of the PCB, and the traces were kept at a 10mil thick width since the board is low power, meaning that it generally requires less than 500 mA. All the components were able to be soldered at WWW Electronics, Inc. (3W) by hand.

Shown below are the final products of the PCB design where Figure 11 displays the final board with the ground plane hidden for more clarity, Figure 12 displays the final board with both power planes visible, and Figure 13 displays the final product as the printed board with all the components soldered onto it.

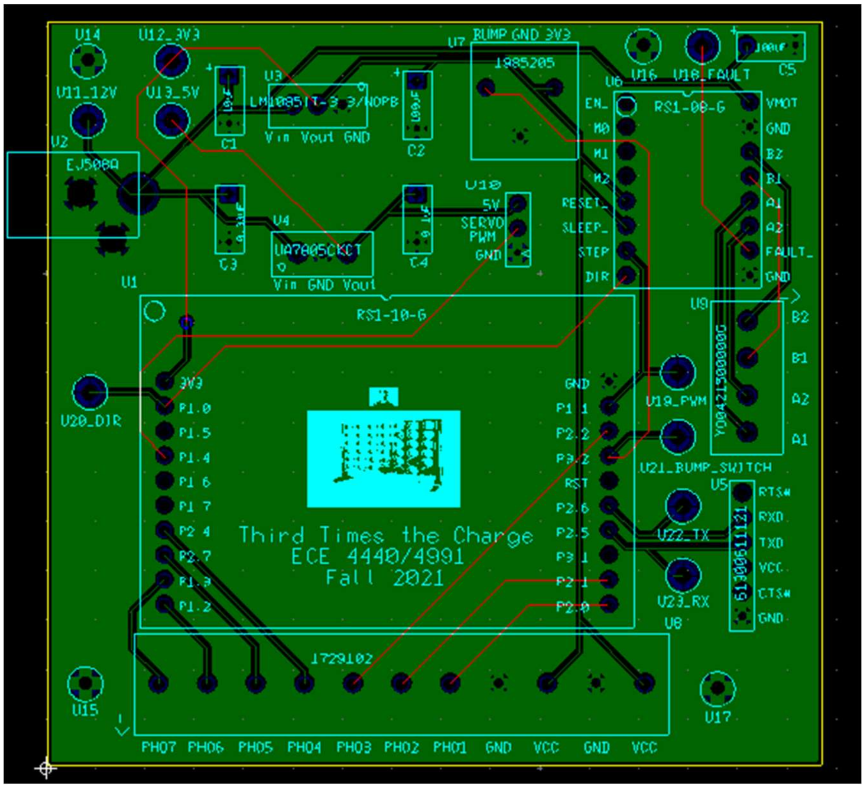

Figure 11. PCB Layout Ground Plane Hidden

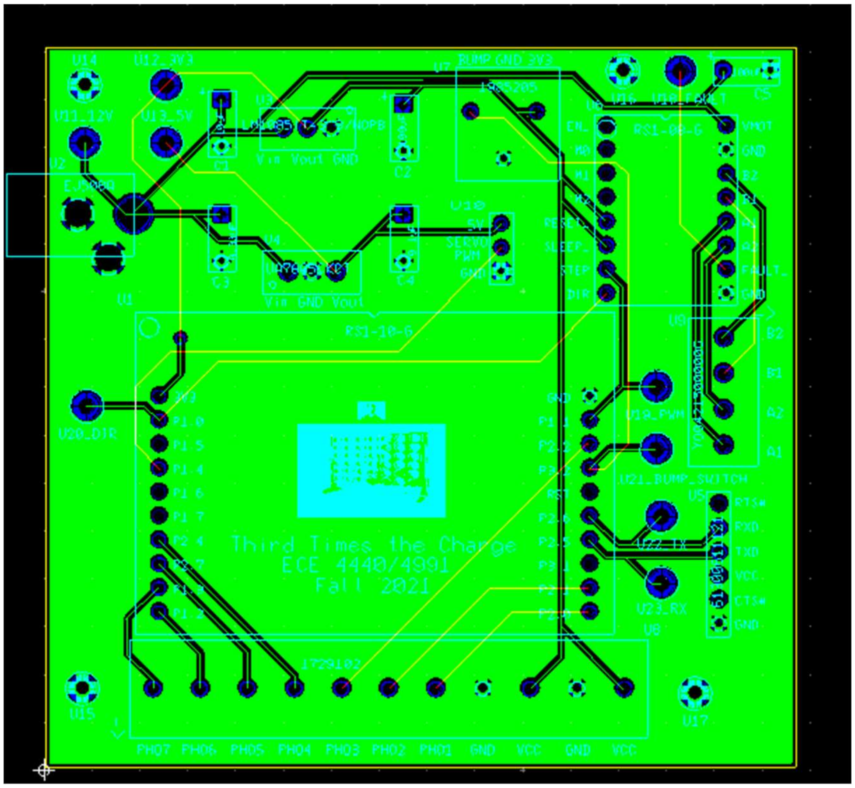

Figure 12. PCB Layout

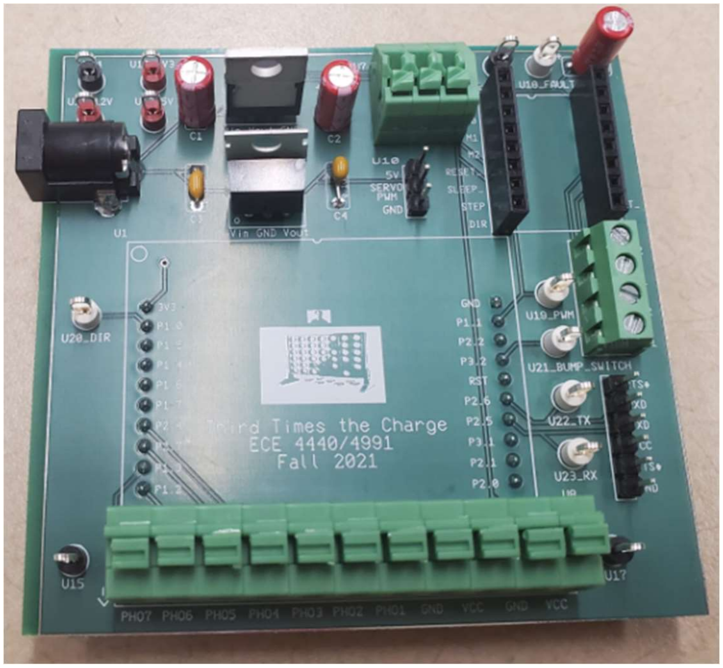

Figure 13. Final PCB Populated

Firmware

The firmware on the MSP primarily acts as the translator between the physical layer of the robot that interacts with the user through motors and sensors and the game algorithm running on the laptop. All the firmware was written in C using Code Composer Studio. A high-level overview of the firmware flow is shown in Figure 14.

The pin mappings for the microcontroller input and output are described in Table 1.

Function	Module	MSP Port.Pin	In/Out
Stepper PWM	TimerA0 PWM	P1.1	Out
Stepper Direction	GPIO	P1.0	Out
Stepper Enable	GPIO	P3.1	Out
Limit Switch	GPIO	P3.2	In
Servo PWM	TimerA1 PWM	P1.4	Out
UART Receive	UART1 RX	P2.5	In
UART Transmit	UART1 TX	P2.6	Out
Photo-Interrupter 1	GPIO	P2.0	In
Photo-Interrupter 2	GPIO	P2.1	In
Photo-Interrupter 3	GPIO	P2.2	In
Photo-Interrupter 4	GPIO	P2.4	In
Photo-Interrupter 5	GPIO	P2.7	In
Photo-Interrupter 6	GPIO	P1.2	In
Photo-Interrupter 7	GPIO	P1.3	In

Table 1. Pin Mappings

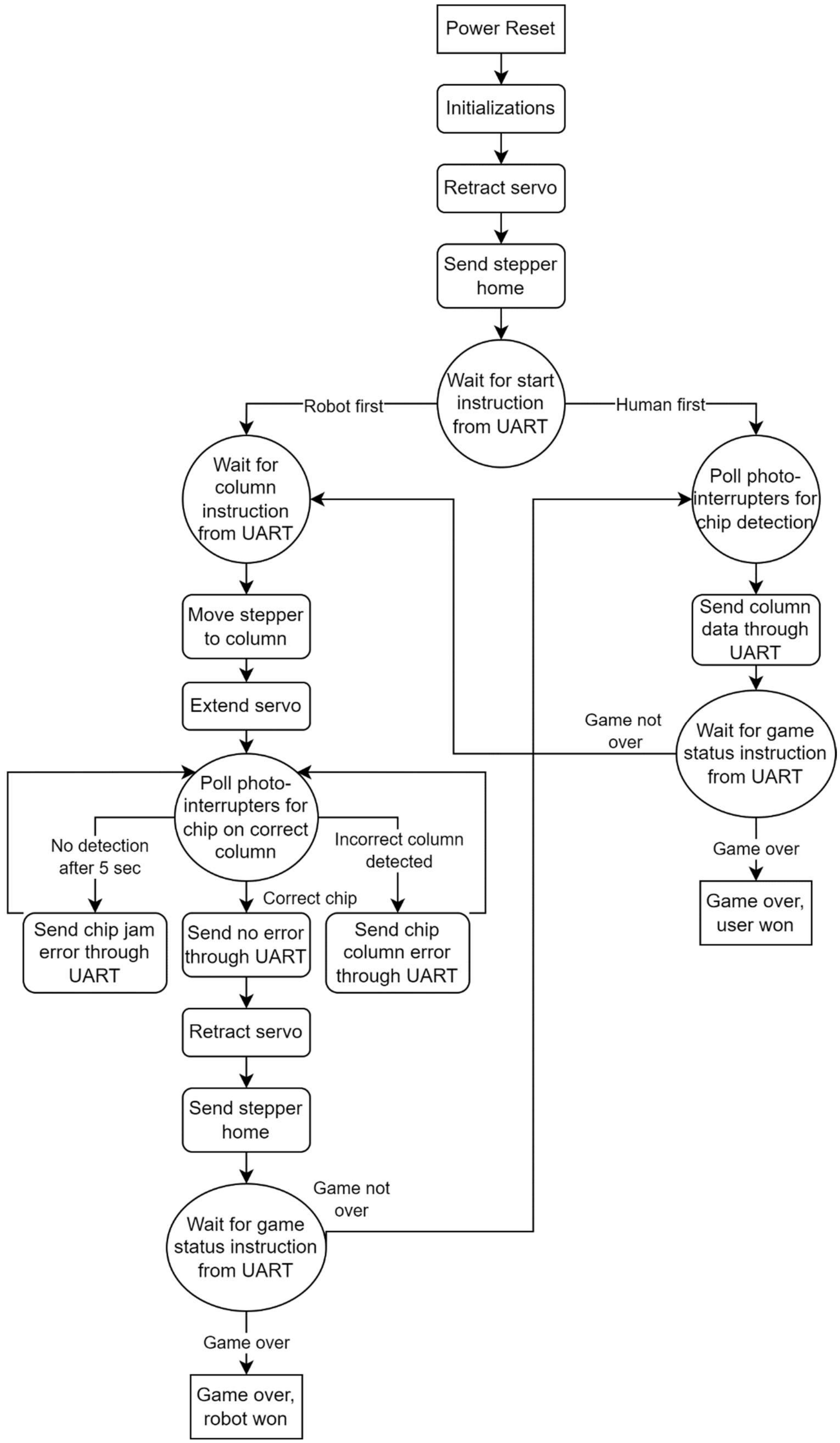

Figure 14. Firmware Overview

Servo Output

The Smraza S51 micro servo is a 180° angle control rotation servo. It is controlled via a PWM signal with high level pulses that last between 500 µs and 2500 µs. Through experimentation, it was determined that high level pulses of 500 µs and 2300 µs are sufficient to drive the gear rack far enough into the chip tube to properly dispense the chips and retract the gear rack far enough so a new chip will fall into queue. With the orientation of the servo in this project, the 500 µs pulses extend the gear rack and the 2300 µs pulses retract the gear rack.

The PWM signal for the servo is generated by TimerA1 on the MSP. Timer1A is configured to source from SMCLK with a divider of 8. SMCLK runs at 2 MHz, so Timer1A has a cycle frequency of 250 kHz. In PWM mode, TimerA1 starts at 0 with the output level high. It counts up until it reaches a match value, at which point the output level goes low. It continues to count until it reaches its maximum value and then resets to 0 and output goes high again. The max value is set as 4999 to give a PWM period of 0.02 s or a frequency of 50 Hz.

$$\frac{MaxValue + 1}{TimerCycleFrequency} = TimerPeriod$$

For the servo extension, the match value at which the output level changes is set to 124 to give a total high time of 500 µs. For the servo retraction, the match value at which the output level changes is set to 574 to give a total high time of 2300 µs.

$$\frac{MatchValue + 1}{TimerCycleFrequency} = HighTime$$

Figure 15 shows the specific functions defined in the servo driver.

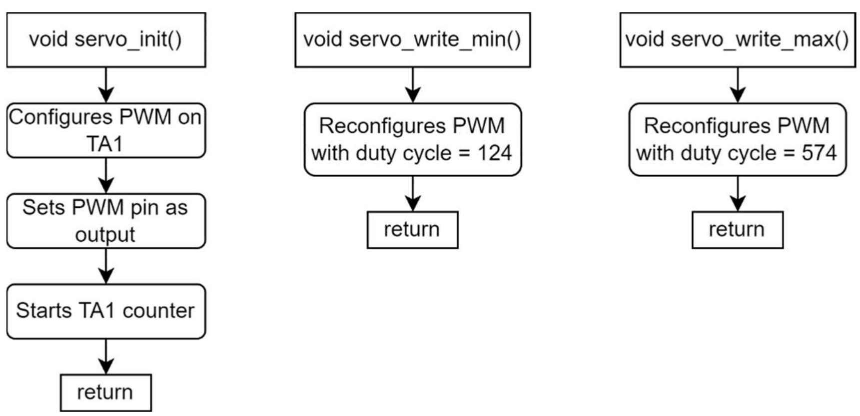

Figure 15. Servo Driver Functions

Stepper Output

The Pololu DRV8825 Stepper Motor Driver Carrier Board takes in logic level PWM signals and outputs high current signals which can drive the NEMA 17 stepper. Each high-level pulse in the PWM signal corresponds to one step driven in the stepper. The signals that need to come from the MSP to the stepper controller are the PWM signal for step control, a binary direction signal which controls the direction the stepper moves, and a binary enable signal that controls whether current will flow into the stepper. The controller also has options for micro-stepping, but that level of precision is not needed in this application, so it was left in full-step mode.

The PWM signal for the stepper is generated by TimerA0 in a very similar manner to the servo PWM signal. TimerA0 is also configured to source from SMCLK with a divider of 8, resulting in a timer cycle frequency of 250 kHz. However, it has a maximum value of 249 and a match value of 124, resulting in a PWM frequency of 1 kHz with a 50% duty cycle. 1 kHz was determined experimentally to be a good frequency that moves the stepper quickly but not too quickly as to cause a safety hazard.

In order to send a specific number of steps, a integer variable *count* is set to the number of steps desired, and TimerA0 is enabled with interrupts. When TimerA0 wraps around to 0, an interrupt is triggered which decrements *count*. When *count* reaches 0, TimerA0 is disabled, and therefore will not send anymore pulses to the stepper controller. The direction can also be controlled with a simple binary variable. When looking on the Connect 4 board side of the robot, direction 0 is left and direction 1 is right. The enable signal is active low, so a 0 is enabled and a 1 is disabled.

In addition, an OpenBuilds Xtension Limit Switch is placed on one side of the chip dispenser which acts a “home” position for the stepper. When the stepper is sent home, the MSP repeatedly sends instructions to move 1 step towards home until the limit switch is activated, at which point the MSP stops sending steps.

Through experimentation, it was determined that 1000 steps in the stepper translate to approximately 141 mm of linear motion on the track.

$$Steps \approx \frac{Distance(mm) \times 1000\ steps}{141\ mm}$$

The Connect 4 board starts approximately 45 mm away from the home position, which corresponds to 319 steps. Each column in the board is approximately 35 mm wide, which corresponds to 248 steps. Column 0 is the column farthest away from the home position and column 6 is the column closest to the home position, so when sending the chip dispenser to a specific column, the MSP sends the number of steps according to the following equation:

$$Steps = 319 + 248(6 - Column)$$

Figure 16 shows the specific functions defined in the stepper driver.

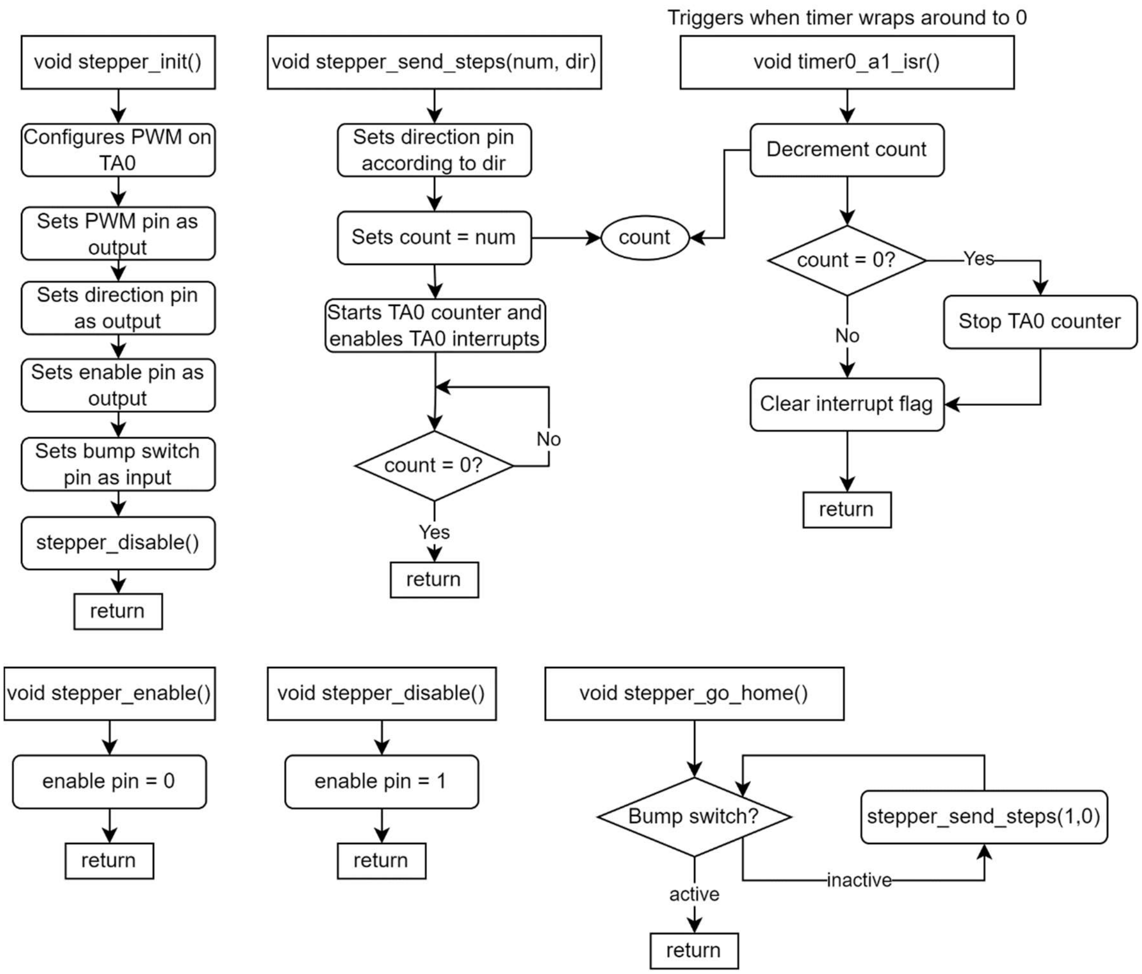

Figure 16. Stepper Driver Functions

Photo-Interrupter Input

The GP1A57HRJ00F photo-interrupters output a logic low signal when an object is obstructing the beam between their detectors and emitters. These outputs are connected directly to GPIO pins on the MSP. Chip detection is done by polling the photo-interrupters and waiting for a change in the received signals.

If after 5 seconds of polling the photo-interrupters, a change is still not detected, and a timeout check is enabled, then a timeout error is returned. This means that the robot's chip got stuck somewhere in the dispensing mechanism. TimerA2 is used to facilitate this error detection. TimerA2 is configured in up mode and is sourced from ACLK with a divider of 64. ACLK runs at 32.768 kHz, so TimerA2 has a cycle frequency of 512 Hz. The desired wait time before a timeout is 5 seconds, so the TimerA2 counter will be equal to 2559 when a timeout is meant to occur.

$$TimeoutCycles = Time\ (seconds) \times TimerCycleFrequency - 1$$

Functions used for the photo-interrupters are shown in Figure 17.

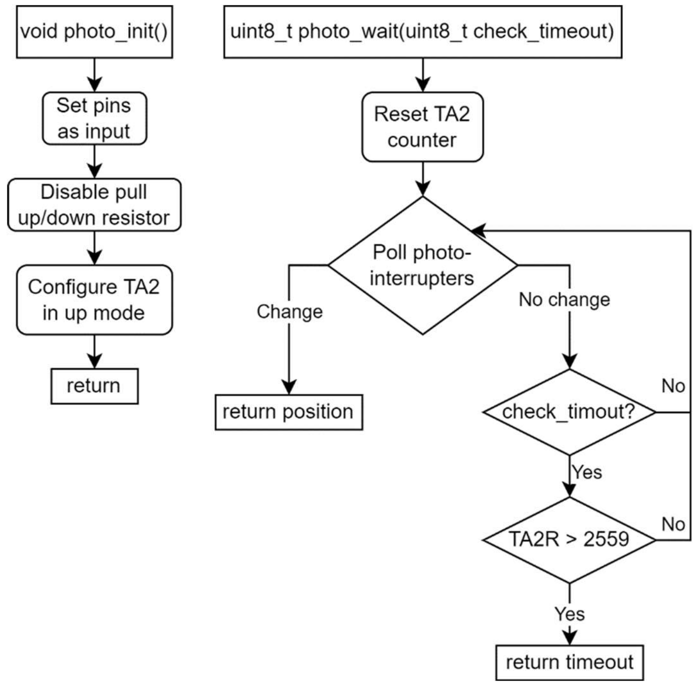

Figure 17. Photo-Interrupter Functions

Communication

A custom serial instruction set is defined that allows the MSP to communicate with the laptop. Each instruction is one byte, including a two-bit header, a three-bit opcode, and a three-bit operand. The instructions are described in Table 2.

Instruction	Opcode Description	Operand Description	ASCII Equivalent
01 000 000	Start Game	Robot First	@
01 000 111	Start Game	Human First	G
01 001 000	Game Status	Not Finished	H
01 001 111	Game Status	Game Over	O
01 101 XYZ	Human Column	XYZ = Column #	h,I,j,k,l,m,n
01 110 XYZ	Robot Column	XYZ = Column #	p,q,r,s,t,u,v
01 111 000	Error	Wrong Column Error	x
01 111 001	Error	Chip Jam Error	y
01 010 111	No Error	No Error	W

Table 22. Serial Instruction Set

The MSP has a UART module built in with a configurable baud rate. This project uses the UART1 module. UART1 is configured to source from SMCLK with a baud rate of 115200, LSB first, one stop bit, and no parity. Additional configuration parameters were determined using a calculator tool [60].

Functions written to send instructions simply encode the data and send it through UART1. Functions written to receive instructions wait until an instruction is received with the proper format and then the data is decoded and returned. An enumerated type called *turn_t* is used to track the status of the game within the firmware. Its values consist of *ROBOT*, *HUMAN*, *TBD*, and *GAME_OVER*. The function used for communication with UART are shown in Figure 18.

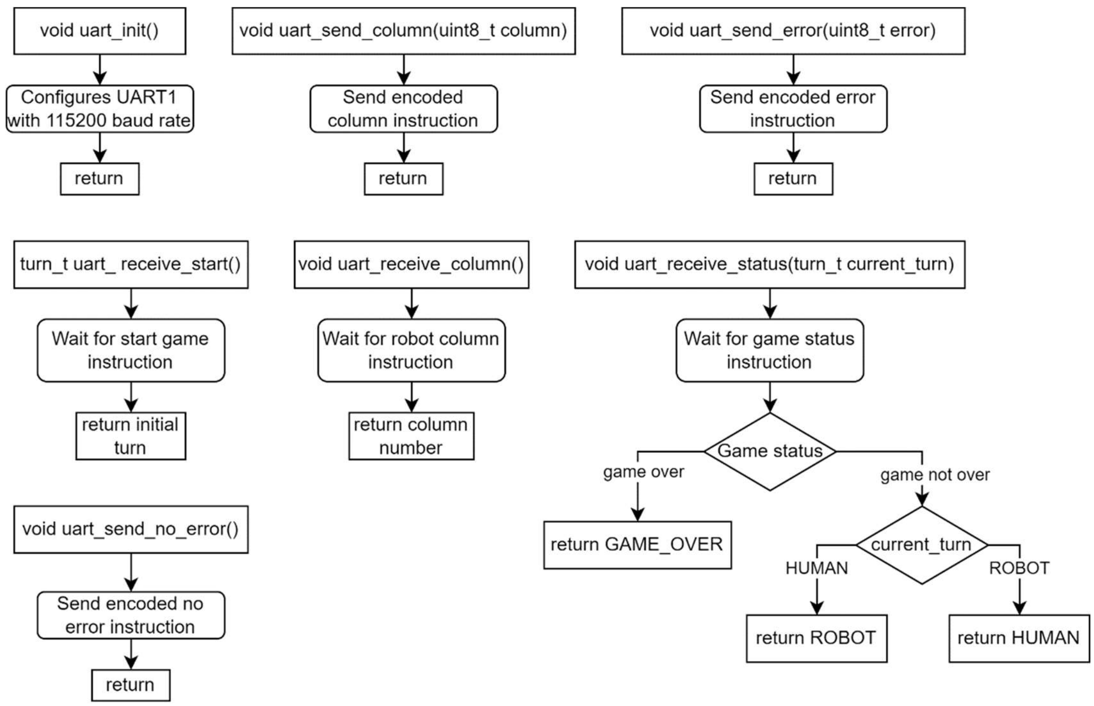

Figure 18. UART Functions

Software

The software for the Connect 4 robotic system acts as the "brain" of the system, allowing for the robot to take in input and produce output for the robot to respond to a human players move. The robot is able to develop a response through the use of a well-known algorithm, known as the minimax algorithm, to produce a move that would allow the robot to win, or prevent the human player from winning the Connect4 game. All of the software that implements the Connect4 game was developed using Java as the main programming language and was coded in Visual Studio Code as the main IDE, or integrated development environment.

Game Algorithm

The game algorithm is used to develop the robotic system's next move. The algorithm implements a commonly used, and well-known algorithm known as the minimax algorithm [61]. The minimax algorithm is used in many artificial intelligence schemes to provide for the best possible scenario for a certain type of response, which in this case is the human players' decision of where to place a Connect 4 chip. The minimax function that is implemented in the game algorithm software is a recursive function that loops down a decision tree of possible moves that can be made when looking at the current state of the tree. The algorithm attempts to maximize its own score while also minimizing the opponents score to develop the next column for which the robot will place a chip in. The function uses a scoring scheme that provides a score for a plethora of scenarios, such as the robot or human having a certain number of chips in a row. This scoring scheme is what allows the game algorithm to know what the next ideal move is for the robot. In order to develop the game algorithm, inspiration was drawn from an independent coder [62] in which the logic that he implemented was translated into a form that would be useful for the game.

Communication

Communication is a key factor in the software the implements the game algorithm. In order for the software that implements the game algorithm to respond to input and produce output, the software utilizes the *jSerialComm* API [31] that defines a SerialPort object the allows for the use of a Serial Communication Port. The SerialPort object is used to read and write bytes via a communication port that is used when the SerialPort object is defined which is shown in Figure 19. Once the Connect4.jar file is executed, the SerialPort is used to write the game instructions to the MSP430 to allow for the firmware to know which player is going first, the human or robot. Depending on which of those is chosen to go first, the software will then wait for input from the SerialPort sent by the MSP430 if the human was chosen to go first or it will produce a column to output and send a command/byte mapped to that column to the MSP430 by writing that byte to the serial communication port. This is the method by which the game implemented in software communicates with the firmware in the MSP430 microcontroller.

```
SerialPort sp = SerialPort.getCommPort("COM6");
sp.openPort();
sp.setBaudRate(115200);
sp.setComPortTimeouts(4096, 10000000, 10000000);
```

Figure 19. SerialPort Initialization

Interface

The interface of the software is handled via the command line. The executable .jar file that is used to run the Connect4 game launches a command line that then allows for the game to be played, asking the user to decide on which player goes first, the human or robot, along with asking for the level of difficulty the player would like to play against, easy or hard. Each of these prompts are responded to by simply responding with “h” or “r” for human or robot going first or “h” or “e” for hard or easy difficulty. From there, the game software outputs the current game board and has the potential to output messages to the user if the robot encounters an error such as a jam in the dispensing mechanism or a chip being placed in the wrong column. Figure 20 displays an example of what the interface of the Connect4 game looks like when executed.

```
------- Welcome to Connect4 -------
Please choose who goes first: Human(h) or Robot(r)
h
Please choose your difficulty: Hard (h) or Easy (e)
h
[0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0]
Here is the current game board
```

Figure 20. Command Line Interface

Project Time Line

GanttProject [63] was used to create the timeline for the project. The original timeline, updated timeline for the Midterm Design Review, and final timeline are shown in Figures 21, 22, and 23. Each timeline took note of important dates in green, such as Fall Break, the Midterm Design Review, Thanksgiving Break, and Demo Day. Each category of tasks, such as Mechanical (blue), PCB (orange), Firmware (magenta), and Software (cyan), were parallelized while subtasks in each category were a combination of parallel and serial. The original timeline was fairly naïve and optimistic, assuming things would get done quickly and on the first try. The update for the Midterm Design Review was more realistic, with updated knowledge of deadlines such as the board sendout dates. It also set a deadline of Thanksgiving to have a mostly working system. For the final timeline, which is what truly occurred throughout the semester, it shows that some things took longer than expected, especially the mechanical design. Many parts had to be redesigned and reprinted throughout the semester because of measuring mistakes or incorrect assumptions. The Thanksgiving deadline was not completely met, but at that point the firmware and mechanical design were working together as expected, and the last step was to integrate the software, which was done soon after Thanksgiving.

The responsibilities were primarily divided up as follows. Kellan led the mechanical designs with support from both Jared and Roman. Jared was primarily responsible for PCB and electronic design with support from Kellan. Kellan was primarily responsible for developing firmware for the microcontroller with support from Roman. Roman was primarily responsible for software development for the PC program with support from Jared. While these were our official roles, due to our small team size, we all contributed to every part of the project in varying amounts.

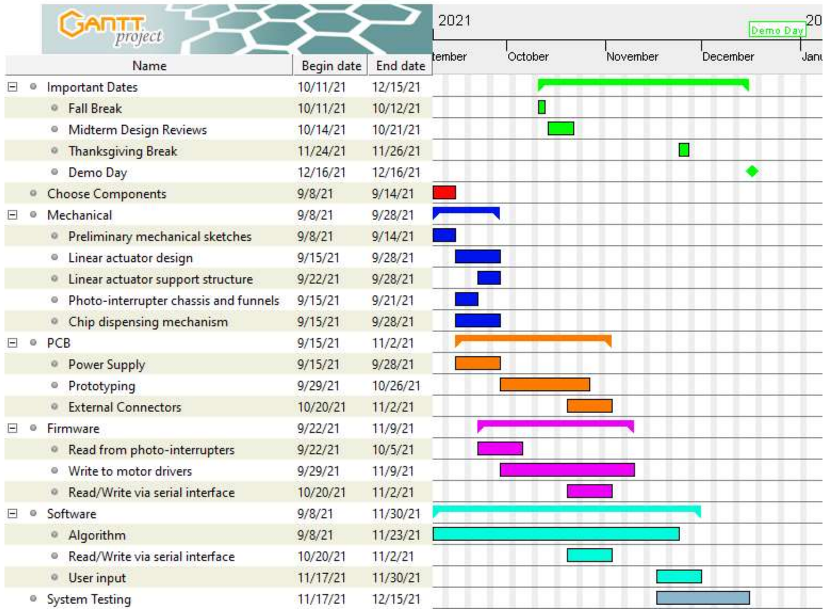

Figure 21. Original Project Timeline

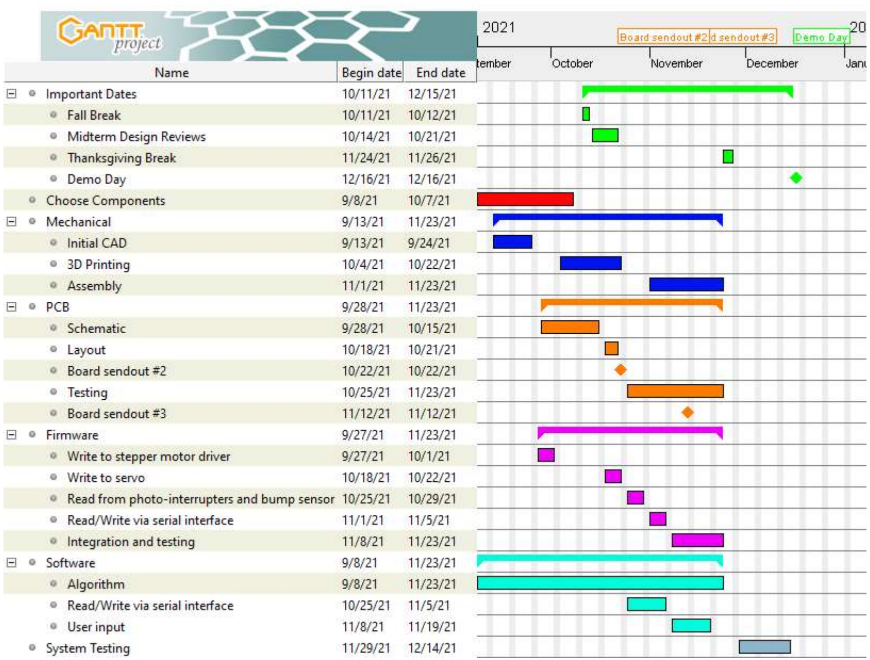

Figure 22. Updated Timeline for Midterm Design Review

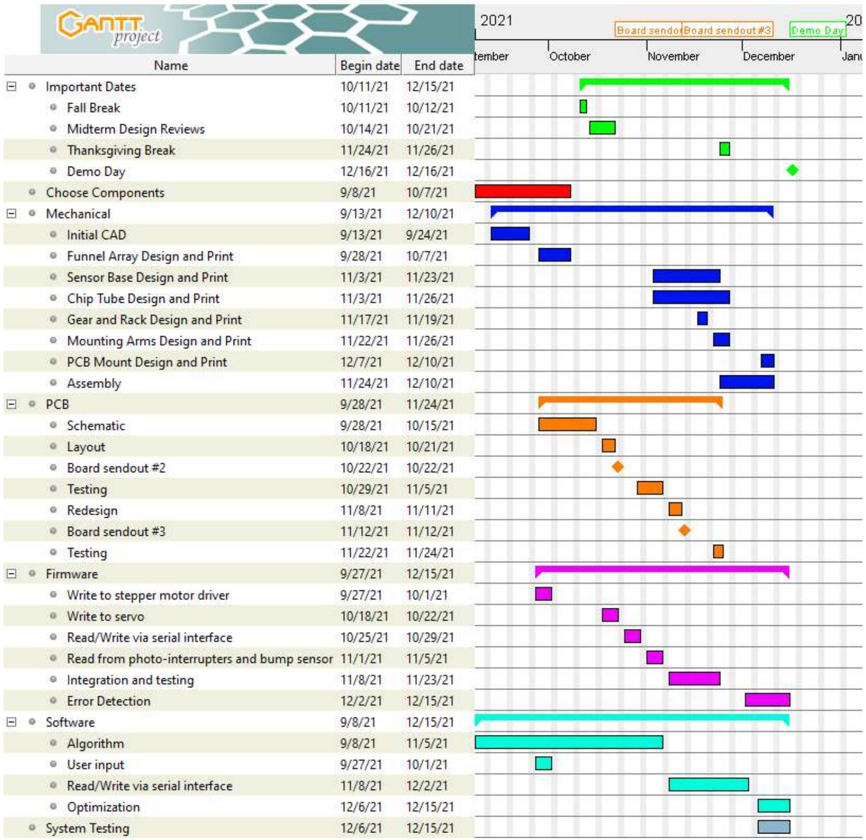

Figure 23. Final Timeline

Test Plan

Figure 24 displays the original test plan made for this project that follows the following path: the whole system was divided into three subsections such that each group member was tasked with a subsection. These subsections were for hardware, software, and firmware with mechanical elements.

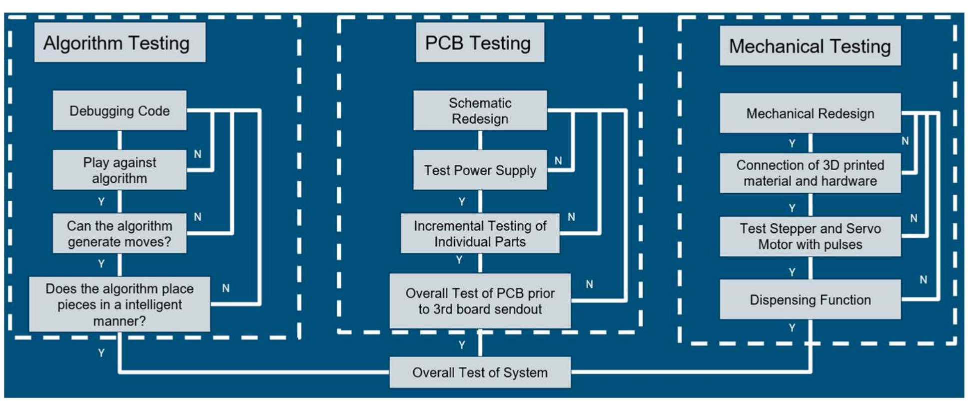

Figure 24. Test Plan

For the software, the testing followed the path of being able to play against a human player. Once this part of the code was working properly, the next step was to generate random moves. Following the test plan, this allowed the code to be refined until it could map out each move made by the human and subsequent moves from the computer with a 6 row by 7 column grid. The final step for the algorithm to successfully work was to generate moves in an intelligent manner utilizing the minimax algorithm which eventually was accomplished, and tests were made with an online Connect 4 algorithm that always plays optimally. The final tests against this other algorithm showed that the project produced algorithm could beat the online algorithm if the project algorithm went first, and the project algorithm could tie if it went second. This test result showed that the project algorithm could reach the optimal state of playing first or second against an optimal algorithm using the minimax algorithm. The algorithm was then tested against human players and results showed that it could thoroughly win against humans a vast majority of the time. Throughout the process of developing the code, if there were ever any debugging issues, the guidelines set out by the original test plan meant that the solution would be to debug the code and ensure that each step still worked when testing to confirm whether the debugging fixed the problem.

For the hardware testing, the plan was followed closely by testing the power supply, the bump switch input, the photo-interrupter input, the stepper output, the servo output, and the UART to USB receive and transmit. For the power supply, each test point was measured to ensure that the values of 3.3V, 5V, and 12V were supplying the expected voltages. The next part was to test the individual subsystems where the bump switch and photo-interrupters both produced 3.3V when not active and 0V when active on their test points. Since the servo motor, the stepper motor, and the USB to UART required some control from the firmware, they were integrated into this step. On the first PCB, the servo motor could move but was stuttered, and from insight into the data sheets the servo motor required more current than was supplied by the 5V regulator, so that led to a change in the components for the regulators to supply up to 1.5A. Also on the first PCB, the stepper controller and the USB to UART modules were not working and nothing for those systems could be tested even with proper inputs from the firmware. This led to another design change where the stepper controller and USB to UART module were simplified to reduce any means of errors from individual components leading to the stepper motor breakout board and the USB to UART cable for a more direct connection between the laptop and MSP. On the final PCB, all these tests were repeated following the PCB test plan and the results showed that each subsystem worked.

For firmware testing, including the mechanical tests, the code was sectioned into different test cases which could be switched by commenting and uncommenting define statements. This included testing for the bump switch input, the photo-interrupter input, the UART input and output, the servo output, the stepper motor output, and for large scale testing of the mechanical system. The microcontroller was successfully able to read the input from the bump switch and photo-interrupters. The UART configuration was initially tested using UART0 being connected to the same cable as debugging which was able to send and receive instructions. When the USB to UART cable arrived, the UART configuration was switched to UART1 and tested with the corresponding pins with positive results as well. The servo PWM output was able to drive the servo motor after adjustments in timing from individual tests. When the stepper controller breakout board arrived, the PWM signal used to drive the stepper motor had a frequency that was too low which caused the stepper motor to make loud noises without any movement. This test resulted in changing the frequency to a higher value which allowed the stepper motor to move in a forward and backward direction. One other important feature found from testing the stepper motor was that the stepper motor would create a loud sound if it was enabled and not moving, and so from this result the correction to this problem was to utilize a GPIO pin from the MSP to enable and disable the stepper motor. With these preliminary steps complete, the mechanical system was able to be tested which included tests through PuTTY to control the chip dispenser and receive instructions when game pieces were placed into the Connect 4 board passing through the photo-interrupters. From more testing the mechanical designs added onto the Connect 4 board were adjusted to optimize the dispensing of the game pieces.

When the whole system was assembled, this allowed for the final stages of testing having integrated the hardware, software, and firmware together. With initial tests playing against the robot, there was a mismatch between the instructions to be sent and received between the firmware and software which allowed for a fix within the code. The only other issue involved the game pieces often getting stuck in the columns, but an adjustment to the chip dispenser fixed this issue.

Final Results

The team was able to complete a fully functional Connect 4 Robot that can play Connect 4 against a human player. The system contains all the initial requirements that the team set out in the final proposal, which was a robotic system that responds to input from a human player and produces output for the human player to respond to on a detachable Connect4 board. The robotic system in its final state allows a player to select whether the human player goes first, or the robot goes first with the ability to also choose a difficulty of easy or hard. The robot is then able to play the game of Connect4 with minimal risk of failure, and in the case of failure, it prompts the user through the command line interface the current issue the robot is facing and how to fix it. The game proceeds until a player, human or robot, has won. When playing on the hard difficulty, the minimax algorithm allows for the robot to win more than 75% of the time with less than 5% chance of chip placement error. With the mechanical design, the Connect 4 board is also able to be attached and detached. With the criteria that were established, and the grading rubric shown in Table YYY, the robotic system is placed in the A+ range. The criteria that were initially set in place are as follows:

- The hardest difficulty poses a challenge to a human player. The robot should win over 75% of the time.
- Robot can detect the user's move without any additional input. There should be zero user input necessary besides them placing their chip in their desired column.
- The robot can detect when the chip is jammed in the dispensing mechanism. After 5 seconds, the user interface should display a message when a chip is jammed.
- The robot can dispense chips into the correct columns without error or user intervention except for a very rare (<5%) chip jam error.

- The user interface on the PC has functioning options for difficulty (hard and easy) and turn order (robot first or user first).
- The user interface on the PC displays the winner when the game is over.
- The Connect 4 board can be attached and detached.

Letter Grade	Criteria
A+	All 7 requirements are met.
A	6 requirements are met.
B+	5 requirements are met.
B	4 requirements are met.
C	2 or 3 requirements are met.
D	0 or 1 requirements are met.

Table 3. Grading Rubric

Although the team was successful in meeting all the goals of the project, there were a few drawbacks that caused the design and implementation of the robotic system to not be ideal/optimal for user experience. One major drawback was the failure to ensure that the device was safe to be around children, especially because of the sharp edges on the metallic frame and exposed electronics and motors. Additionally, although there is a mount for the microcontroller and PCB, there is still risk of coming into contact with exposed wires, which was not what was initially planned for but due to timing constraints could not be addressed.

Another factor that was not set as an expectation, but a goal, was a more interactive user interface that implemented a java GUI or Jframe. This would have been a better method of interfacing the computer, however for the sake of time and ensuring that the program was ready for demonstration at the Capstone Fair, the creation of a more interactive interface was set aside.

The final results have shown the team that meeting initial requirements are of the utmost importance and we must schedule our time in order to properly meet those goals prior to attempting to add additional features that simply improve on the design rather than being a core feature of the design.

Costs

Total project costs came to $402.34, as outlined in Table 4. This falls well within the project budget of $500.

Item	Total Cost
Electronic Parts	$208.27
Mechanical Parts	$73.27
PCB Manufacturing	$66.00
PCB Assembly 1	$37.80
PCB Assembly 2	$17.00
Total Cost	$402.34

Table 44. Total Project Costs

A couple notable items were acquired with no cost to the team because they available for use from the NI Lab or from the team's personal collection. This includes the NEMA 17 stepper and the linear rail on which it is attached to a belt and cart, the Smraza servo, as well as the Connect 4 board itself and the chips. The 3D printed parts were also produced at no cost thanks to the UVA 3D Printing Studio. Several electronic parts were also ordered intended to be put on the second PCB, but then the PCB was drastically redesigned without the need for many of those parts, so those parts and money were wasted.

The cost of one entire Connect 4 Robot if it were to be built with the current design including the parts that were acquired at no charge is $252.29. A breakdown of costs at limited quantities and in high quantities is shown in Table 5, with more detailed costs shown in Appendix B.

Item	Cost per unit for low quantities	Cost per unit for 10,000 produced
Electronic Parts	$79.77	$70.92
Mechanical Parts	$122.52	$122.52
PCB Manufacturing	$33.00	$6.04*
PCB Assembly	$17.00	$17.00
Total Cost	$252.29	$216.48

Table 55. Costs to Produce One Connect 4 Robot

*According to Advanced Circuits Custom Quote [64]

The most expensive electronic parts are the USB to UART cable, the stepper controller carrier board, and the NEMA 17 stepper. None of these parts had bulk discounts on the vendors, so the increased cost would continue to pile up when producing high quantities. The most expensive mechanical parts are the ones that are needed to build the cart that travels horizontally on the linear rail. The vendor that was used to supply mechanical parts does not offer bulk discounts, so if this were to be manufactured in bulk, another vendor should be explored. With more sources of bulk discounts, the cost could be decreased dramatically at higher quantities. An automated PCB assembly process should also be explored.

Future Work

To improve upon the current version of the Connect 4 Robot, the team suggests improving the safety of the system, decreasing costs, and designing a better user experience. The following were all goals that the team wanted to fulfill in this project seeks to fulfill, but they ended up being a little too much to handle in one semester. With the knowledge gained and reported for this project, future teams would be able to accomplish them.

To increase the safety of the system, future teams should use a NEMA Type 1 enclosure [65] to house the PCB and electronics. This would protect the electronics from accidental touches, dust, and indirect splashing. There should also be some way of either blocking the user from touching the motors or detecting when they get touched and stopping them. This would make the product much safer to use and much more child friendly.

One way to assuredly decrease the cost of the electronic parts are to implement the stepper driver and USB-UART functionality into the PCB instead of having a separate carrier board and cable. There are many available integrated circuits for driving stepper motors and bridging USB and UART, such as the DRV8825 [REFERENCE] and FT231X [REFERENCE]. Putting these directly on a board along with their necessary passive components would be a much cheaper alternative to the separate modules. Additionally, SMD components could be explored to further reduce the costs. Another way to reduce costs would be to change the mechanical design of the system. This is primarily an electrical and computer engineering project, so mechanical optimization is not necessary the goal, but other materials besides extruded aluminum may prove to be cheaper and/or easier to use.

Finally, the user interface on the laptop could be improved to provide a more intuitive user experience. Most consumers are not used to interacting with a program through a terminal window, so a graphical user interface could improve their experience. It could include buttons to select difficulty and turn in order instead of typing in commands. It could also display a live, graphical representation of the Connect 4 board that is easier to understand than black and white characters.

References

[1] "Connect 4 Game," *Shop.hasbro.com*. [Online]. Available: https://shop.hasbro.com/en-us/product/the-classic-game-of-connect-4:80FB5BCA-5056-9047-F5F4-5EB5DF88DAF4. [Accessed: 16-Dec-2021].

[2] D. Edwards, "Pandemic lockdowns boost consumer robotics market to $10 billion in 2020 revenues," *Robotics & Automation News*, 16-Feb-2021. [Online]. Available: https://roboticsandautomationnews.com/2021/02/16/pandemic-lockdowns-boost-consumer-robotics-market-to-10-billion-in-2020-revenues/40492/. [Accessed: 16-Dec-2021].

[3] "Understanding the effects of social isolation on Mental Health," *Online Public Health & Healthcare Administration Degrees*, 08-Dec-2020. [Online]. Available: https://publichealth.tulane.edu/blog/effects-of-social-isolation-on-mental-health/. [Accessed: 16-Dec-2021].

[4] P. McCabe, "Connect Four Robot," *Patrick McCabe Makes*. [Online]. Available: http://patrickmccabemakes.com/hardware/Connect_Four_Robot/. [Accessed: 16-Dec-2021].

[5] "MSP430FR2433," *MSP430FR2433 data sheet, product information and support | TI.com*. [Online]. Available: https://www.ti.com/product/MSP430FR2433. [Accessed: 16-Dec-2021].

[6] "MSP-EXP430FR2433," *MSP-EXP430FR2433 Development kit | TI.com*. [Online]. Available: https://www.ti.com/tool/MSP-EXP430FR2433. [Accessed: 16-Dec-2021].

[7] "Multisim download," *NI*. [Online]. Available: https://www.ni.com/en-us/support/downloads/software-products/download.multisim.html#312060. [Accessed: 16-Dec-2021].

[8] "Ultiboard download," *NI*. [Online]. Available: https://www.ni.com/en-us/support/downloads/software-products/download.ultiboard.html#311427. [Accessed: 16-Dec-2021].

[9] "CCSTUDIO," *CCSTUDIO IDE, configuration, compiler or debugger | TI.com*. [Online]. Available: https://www.ti.com/tool/CCSTUDIO. [Accessed: 16-Dec-2021].

[10] "Visual studio code – code editing. Redefined," 03-Nov-2021. [Online]. Available: https://code.visualstudio.com/. [Accessed: 16-Dec-2021].

[11] "Tinkercad." [Online]. Available: https://www.tinkercad.com/dashboard. [Accessed: 16-Dec-2021].

[12] “MakerBot print software – free 3D printing software,” *MakerBot*. [Online]. Available: https://www.makerbot.com/3d-printers/apps/makerbot-print/. [Accessed: 16-Dec-2021].

[13] “Advanced circuits,” *2 Layer & 4 Layer PCB Prototyping | Advanced Circuits*. [Online]. Available: https://www.4pcb.com/pcb-prototype-2-4-layer-boards-specials.html. [Accessed: 16-Dec-2021].

[14] “DigiKey electronics – electronic components distributor,” *DigiKey Electronics – Electronic Components Distributor*. [Online]. Available: https://www.digikey.com/. [Accessed: 16-Dec-2021].

[15] “OpenBuilds,” *OpenBuilds Part Store*. [Online]. Available: https://openbuildspartstore.com/. [Accessed: 16-Dec-2021].

[16] “Small parts for toys and children’s products business guidance,” *U.S. Consumer Product Safety Commission*. [Online]. Available: https://www.cpsc.gov/Business--Manufacturing/Business-Education/Business-Guidance/Small-Parts-for-Toys-and-Childrens-Products. [Accessed: 16-Dec-2021].

[17] M. Barr, Embedded c coding standard. Germantown, MD: Barr Group, 2018.

[18] J. G. Stallcup, “Class 1 circuits,” *Electrical Contractor Magazine*. [Online]. Available: https://www.ecmag.com/section/codes-standards/class-1-circuits. [Accessed: 16-Dec-2021].

[19] “Generic Standard on Printed Board Design – IPC.” [Online]. Available: https://www.ipc.org/TOC/IPC-2221A.pdf. [Accessed: 16-Dec-2021].

[20] “Specification for base materials for rigid and multilayer printed circuit boards – IPC.” [Online]. Available: https://www.ipc.org/TOC/IPC-4101C.pdf. [Accessed: 16-Dec-2021].

[21] “IPC-A-600J: Acceptability of printed boards table of contents.” [Online]. Available: https://www.ipc.org/TOC/IPC-A-600J.pdf. [Accessed: 16-Dec-2021].

[22] “RoHS Guide,” *2022 RoHS Compliance Guide: Regulations, 10 Substances, Exemptions*. [Online]. Available: https://www.rohsguide.com/. [Accessed: 16-Dec-2021].

[23] “Serial communication.” [Online]. Available: https://learn.sparkfun.com/tutorials/serial-communication. [Accessed: 16-Dec-2021].

[24] “USB standards: USB 1, USB 2, USB 3, USB 4 - capabilities & comparisons,” *Electronics Notes*. [Online]. Available: https://www.electronics-notes.com/articles/connectivity/usb-universal-serial-bus/standards.php. [Accessed: 16-Dec-2021].

[25] “Replicator+ desktop 3D printer,” *MakerBot*. [Online]. Available: https://www.makerbot.com/3d-printers/replicator/. [Accessed: 16-Dec-2021].

[26] "RMC 3D Printing Studio," *Uva Library*. [Online]. Available: https://www.library.virginia.edu/services/3d-printing/. [Accessed: 16-Dec-2021].

[27] "MSPDRIVERLIB," *MSPDRIVERLIB Driver or library | TI.com*. [Online]. Available: https://www.ti.com/tool/MSPDRIVERLIB#downloads. [Accessed: 16-Dec-2021].

[28] "stdint.h." [Online]. Available: https://pubs.opengroup.org/onlinepubs/009604599/basedefs/stdint.h.html. [Accessed: 16-Dec-2021].

[29] "Package java.util," *java.util (Java Platform SE 8)*, 11-Oct-2021. [Online]. Available: https://docs.oracle.com/javase/8/docs/api/java/util/package-summary.html. [Accessed: 16-Dec-2021].

[30] "Package java.io," *java.io (Java Platform SE 8)*, 11-Oct-2021. [Online]. Available: https://docs.oracle.com/javase/8/docs/api/java/io/package-summary.html. [Accessed: 16-Dec-2021].

[31] "JSerialComm." [Online]. Available: https://fazecast.github.io/jSerialComm/. [Accessed: 16-Dec-2021].

[32] "Metals (other than cans) recycling," *Less Is More*, 13-May-2020. [Online]. Available: https://lessismore.org/materials/14-metal/. [Accessed: 16-Dec-2021].

[33] "Command large picture hanging strips | 3m United States." [Online]. Available: https://www.command.com/3M/en_US/command/products/~/Command-Large-Picture-Hanging-Strips/?N=5924736+3291751934+3294529207&rt=rud. [Accessed: 16-Dec-2021].

[34] L. T. Jones, A. Howden, M. S. Knighton, A. Sims, D. L. Kittinger, and R. E. Hollander, "US4398720A - Robot Computer Chess Game," *Google Patents*, 05-Jan-1981. [Online]. Available: https://patents.google.com/patent/US4398720A/en. [Accessed: 16-Dec-2021].

[35] M. M. Maharbiz and S. Jaqua, "US10953314B2 - intelligent game system for putting intelligence into board and tabletop games including miniatures," *Google Patents*, 03-Jun-2008. [Online]. Available: https://patents.google.com/patent/US10953314B2/en?q=intelligent%2Bgame%2Bplay&oq=intelligent%2Bgame%2Bplay. [Accessed: 16-Dec-2021].

[36] K. Nakazumi, E. Nishida, T. Ohno, and M. Ito, "US8261904B2 - coin depositing and dispensing machine," *Google Patents*, 01-Feb-2007. [Online]. Available: https://patents.google.com/patent/US8261904B2/en?q=coin%2Bejection&oq=coin%2Bejection. [Accessed: 16-Dec-2021].

[37] "Nema 17 Stepper Motor," *OpenBuilds Part Store*. [Online]. Available: https://openbuildspartstore.com/nema-17-stepper-motor/. [Accessed: 16-Dec-2021].

[38] “Smraza 10 pcs SG90 9G Micro Servo Motor Kit for RC Robot Arm Helicopter Airplane Car Boat Control, Arduino Project-S51,” *smraza*. [Online]. Available: https://www.smraza.com/collections/robotics-cnc-servos/products/smraza-10-pcs-sg90-9g-micro-servo-motor-kit-for-rc-robot-arm-helicopter-airplane-car-boat-control-arduino-project-s51. [Accessed: 16-Dec-2021].

[39] “Loctite epoxy instant mix 5 minute from loctite adhesives,” *Loctite® Epoxy Instant Mix™ 5 Minute*. [Online]. Available: https://www.loctiteproducts.com/en/products/build/epoxies/loctite_epoxy_instantmix5minute.html. [Accessed: 16-Dec-2021].

[40] “GP1A57HRJ00F,” *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/sharp-socle-technology/GP1A57HRJ00F/718287. [Accessed: 16-Dec-2021].

[41] “Linear Rail - V-Slot® Linear Rail,” *OpenBuilds Part Store*. [Online]. Available: https://openbuildspartstore.com/v-slot-linear-rail-1/. [Accessed: 16-Dec-2021].

[42] “RS1-10-G,” *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/adam-tech/RS1-10-G/9832059. [Accessed: 16-Dec-2021].

[43] “5012,” *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/keystone-electronics/5012/255334. [Accessed: 16-Dec-2021].

[44] “5011,” *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/keystone-electronics/5011/255333. [Accessed: 16-Dec-2021].

[45] “61300311121,” *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/w%C3%BCrth-elektronik/61300311121/4846825. [Accessed: 16-Dec-2021].

[46] “Yo0421500000g,” *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/amphenol-anytek/YO0421500000G/2261308. [Accessed: 16-Dec-2021].

[47] “GSM60U12-P1J,” *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/mean-well-usa-inc/GSM60U12-P1J/9593325?s=N4IgTCBcDaIIwA4BsSC0BWdYAsqByAIiALoC%2BQA. [Accessed: 16-Dec-2021].

[48] “EJ508A,” *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/mpd-memory-protection-devices/EJ508A/2439547. [Accessed: 16-Dec-2021].

[49] "UA7805CKCT," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/UA7805CKCT/3458146. [Accessed: 16-Dec-2021].

[50] "LM1085IT-3.3/NOPB," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/LM1085IT-3-3-NOPB/363563. [Accessed: 16-Dec-2021].

[51] "5010," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/keystone-electronics/5010/255332. [Accessed: 16-Dec-2021].

[52] "61300611121," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/w%C3%BCrth-elektronik/61300611121/4846833. [Accessed: 16-Dec-2021].

[53] "TTL-234X-3v3," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/ftdi-future-technology-devices-international-ltd/ttl-234x-3v3/6823715. [Accessed: 16-Dec-2021].

[54] "drv8825 Stepper Motor Driver Carrier," *Pololu Robotics & Electronics*. [Online]. Available: https://www.pololu.com/product/2133. [Accessed: 16-Dec-2021].

[55] "RS1-08-G," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/adam-tech/RS1-08-G/9832056. [Accessed: 16-Dec-2021].

[56] "1985205," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/phoenix-contact/1985205/568660. [Accessed: 16-Dec-2021].

[57] "Xtension Limit Switch Kit," *OpenBuilds Part Store*. [Online]. Available: https://openbuildspartstore.com/xtension-limit-switch-kit/. [Accessed: 16-Dec-2021].

[58] "1729102," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/phoenix-contact/1729102/260613. [Accessed: 16-Dec-2021].

[59] "GP1A57HRJ00F," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/sharp-socle-technology/GP1A57HRJ00F/718287. [Accessed: 16-Dec-2021].

[60] "MSP430 USCI/EUSCI UART baud rate calculation." [Online]. Available: http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSP430BaudRateConverter/index.html. [Accessed: 16-Dec-2021].

[61] Baeldung, "Introduction to minimax algorithm with a java implementation," *Baeldung*, 14-Dec-2021. [Online]. Available: https://www.baeldung.com/java-minimax-algorithm. [Accessed: 16-Dec-2021]. "Free Project Management Tool for Windows, macos and linux," *GanttProject*. [Online]. Available: https://www.ganttproject.biz/. [Accessed: 16-Dec-2021].

[62] K. Galli, "How to program a Connect 4 ai (implementing the minimax algorithm)," *YouTube*, 02-Jan-2019. [Online]. Available: https://www.youtube.com/watch?v=MMLtza3CZFM&ab_channel=KeithGalli. [Accessed: 16-Dec-2021].

[63] "Free Project Management Tool for Windows, macos and linux," *GanttProject*. [Online]. Available: https://www.ganttproject.biz/. [Accessed: 16-Dec-2021].

[64] "Start Instant PCB Quote," *Printed Circuit Board Manufacturing & Assembly Quotes | Advanced Circuits*. [Online]. Available: https://www.4pcb.com/instant_quote/index.html. [Accessed: 16-Dec-2021].

[65] "Nema and IEC Enclosure Classifications," *Engineering ToolBox*. [Online]. Available: https://www.engineeringtoolbox.com/nema-iec-enclosure-standards-d_920.html. [Accessed: 16-Dec-2021].

[66] "DRV8825PWP," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/DRV8825PWP/2676775. [Accessed: 16-Dec-2021].

[67] "FT231XS-U," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/ftdi-future-technology-devices-international-ltd/FT231XS-U/3749471. [Accessed: 16-Dec-2021].

Appendix A: Design Files and Code Links

Mechanical Designs:

https://drive.google.com/file/d/1JRW8MW25Lj0cQMhPdso_3PTQxFVQePfb/view?usp=sharing

Schematics Files:

https://drive.google.com/drive/folders/1U6vbkXVSvFQd6JtBmEkStQ_IkFbBwebr?usp=sharing

PCB Design File:

https://drive.google.com/file/d/1O5SKfENgdJreFhL8yrOBZHCu0DckF5SN/view?usp=sharing

Firmware Code:

https://github.com/KellanED/connect4_control

Software Code:

https://github.com/rpk3xd/connect4

Appendix B: Costs

Quantity	Description	Vendor	Unit Price	Total Price
1	MSP430FR2433 LaunchPad	Digikey	11.99	11.99
12	Test Point, White	Digikey	0.42	5.04
10	Test Point, Red	Digikey	0.42	4.2
10	Test Point, Black	Digikey	0.42	4.2
2	Terminal Block, 3P	Digikey	0.51	1.02
1	Terminal Block, 4P	Digikey	1.17	1.17
2	Terminal Block, 11P	Digikey	1.93	3.86
2	Receptable Strip, 8P	Digikey	0.52	1.04
4	Receptable Strip, 10P	Digikey	0.65	2.6
2	Vertical Header, 3P	Digikey	0.13	0.26
1	Vertical Header, 6P	Digikey	0.34	0.34
2	Power Jack	Digikey	1.33	2.66
2	Micro USB Connector	Digikey	1.71	3.42
7	Photo-Interrupters	Digikey	3.91	27.37
1	AC/DC Wall Mount Adapter, 12V 24W	Digikey	14.94	14.94
1	AC/DC Wall Mount Adapter, 12V 54W	Digikey	35.96	35.96
1	Linear Regulator, 3.3V 100mA	Digikey	2.63	2.63
1	Linear Regulator, 3.3V 3A	Digikey	2.46	2.46
1	Linear Regulator, 5V 300mA	Digikey	2.66	2.66
1	Linear Regulator, 5V 1.5A	Digikey	0.9	0.9
2	CP2103 USB-UART Bridge	Digikey	4.88	9.76
1	FTDI USB-UART Cable	Digikey	21.75	21.75
2	DRV8824 Stepper Motor Driver	Digikey	4.9	9.8
1	DRV8825 Stepper Motor Driver Carrier	Pololu	13.49	13.49
16	TVS Diode, 3.3V	Digikey	0.7	11.2
6	TVS Diode, 5	Digikey	0.62	3.72
5	Resistor, 0.4 Ω, 0603	Digikey	0.31	1.55
5	Resistor, 4.7kΩ, 1206	Digikey	0.1	0.5
5	Resistor, 10kΩ, 1206	Digikey	0.1	0.5
5	Resistor, 30kΩ, TH	Digikey	0.1	0.5
1	Capacitor, Ceramic, 0.1μF, TH	Digikey	0.23	0.23
5	Capacitor, Ceramic, 0.1μF, 0201	Digikey	0.1	0.5
1	Capacitor, Ceramic, 0.33μF, TH	Digikey	0.32	0.32
5	Capacitor, Ceramic, 1μF, 0201	Digikey	0.1	0.5
1	Capacitor, Electrolytic, 10μF, TH	Digikey	0.1	0.1
2	Capacitor, Electrolytic, 100μF, TH	Digikey	0.11	0.22
1	Xtension Connector Set, 4 Pin	OpenBuilds	3.49	3.49
1	Xtension Wire By the Foot, 3 Conductor	OpenBuilds	0.61	0.61
1	Xtension Wire By the Foot, 4 Conductor	OpenBuilds	0.81	0.81
			Total Price	208.27

Figure 25. Total Project Costs for Electronics

Quantity	Description	Vendor	Unit Price	Total Price
2	V-Slot 20x20 Linear Rail, 250mm	OpenBuilds	3.29	6.58
3	V-Slot 20x40 Linear Rail, 500mm	OpenBuilds	6.99	20.97
4	Low Profile Screws M5, 10 Pack, 8mm	OpenBuilds	0.99	3.96
4	Tee Nuts, M5, 10 Pack	OpenBuilds	2.99	11.96
20	Cast Corner Bracket	OpenBuilds	1.49	29.8
			Total Price	73.27

Figure 26. Total Project Costs for Mechanical Parts

Quantity	Description	Vendor	Manufacturer	Manufacturer Part No.	Vendor Part No.	Unit Price	Total Price	Bulk Unit Price	Bulk Total Price
6	Test Point, White	Digikey	Keystone Electronics	5012	36-5012-ND	0.42	2.52	0.17044	1.02
3	Test Point, Red	Digikey	Keystone Electronics	5010	36-5010-ND	0.42	1.26	0.17044	0.51
4	Test Point, Black	Digikey	Keystone Electronics	5011	36-5011-ND	0.42	1.68	0.17044	0.68
1	Terminal Block, 3P	Digikey	Phoenix Contact	1985205	277-1623-ND	0.51	0.51	0.37128	0.37
1	Terminal Block, 4P	Digikey	Amphenol Anytek	YO0421500000G	609-3920-ND	1.17	1.17	0.4102	0.41
1	Terminal Block, 11P	Digikey	Phoenix Contact	1792957	277-2527-ND	1.93	1.93	1.666	1.67
2	Receptacle Strip, 8P	Digikey	Adam Tech	RS1-08-G	2057-RS1-08-G-ND	0.52	1.04	0.2183	0.44
2	Receptable Strip 10P	Digikey	Adam Tech	RS1-10-G	2057-RS1-10-G-ND	0.65	1.3	0.27287	0.55
1	Vertical Header, 3P	Digikey	Würth Elektronik	61300311121	732-5316-ND	0.13	0.13	0.063	0.06
1	Vertical Header, 6P	Digikey	Würth Elektronik	61300611121	732-5319-ND	0.34	0.34	0.166	0.17
1	Power Jack	Digikey	MPD	EJ508A	EJ508A-ND	1.33	1.33	0.66555	0.67
1	Linear Regulator, 3.3V 3A	Digikey	Texas Instruments	LM1085IT-3.3/NOPB	LM1085IT-3.3/NOPB-ND	2.46	2.46	1.379	1.38
1	Linear Regulator, 5V 1.5A	Digikey	Texas Instruments	UA7805CKCT	296-39515-5-ND	0.9	0.9	0.39025	0.39
1	Capacitor, Ceramic, 0.1µF, TH	Digikey	KEMET	C315C104M5U5TA	399-4151-ND	0.23	0.23	0.0418	0.04
1	Capacitor, Ceramic, 0.33µF, TH	Digikey	KEMET	C412C334M5U5TA	399-17284-ND	0.32	0.32	0.06716	0.07
1	Capacitor, Electrolytic, 10µF, TH	Digikey	Würth Elektronik	860010372001	732-8593-1-ND	0.1	0.1	0.054	0.05
2	Capacitor, Electrolytic, 100µF, TH	Digikey	Würth Elektronik	860010372006	732-8598-1-ND	0.11	0.22	0.058	0.12
1	FTDI USB-UART Cable	Digikey	FTDI	TTL-234X-3V3	768-1319-ND	21.75	21.75	21.75	21.75
1	DRV8825 Stepper Motor Driver Carrier	Pololu	Pololu	2982	2982	13.49	13.49	13.49	13.49
1	Xtension Wire By the Foot, 3 Conductor	OpenBuilds	OpenBuilds	2550-By-the-Foot	2550-By-the-Foot	0.61	0.61	0.61	0.61
1	Xtension Limit Switch Kit	OpenBuilds	OpenBuilds	2805-Kit	2805-Kit	6.29	6.29	6.29	6.29
1	NEMA 17 Stepper Motor	OpenBuilds	OpenBuilds	623	623	17.99	17.99	17.99	17.99
1	Smraza Micro Servo Motor	Amazon	Smraza	SG90	B07L2SF3R4	2.2	2.2	2.2	2.20
2	V-Slot 20x20 Linear Rail, 250mm	OpenBuilds	OpenBuilds	280-LP	280-LP	3.29	6.58	3.29	6.58
4	V-Slot 20x40 Linear Rail, 500mm	OpenBuilds	OpenBuilds	155-LP	155-LP	6.99	27.96	6.99	27.96
4	Low Profile Screws M5, 10 Pack, 8mm	OpenBuilds	OpenBuilds	946-pack	946-pack	0.99	3.96	0.99	3.96
4	Tee Nuts, M5, 10 Pack	OpenBuilds	OpenBuilds	536-pack	536-pack	2.99	11.96	2.99	11.96
20	Cast Corner Bracket	OpenBuilds	OpenBuilds	490	490	1.49	29.8	1.49	29.8
1	OpenRail Gantry Kit, 40mm	OpenBuilds	OpenBuilds	2701-Set	2701-Set	31.99	31.99	31.99	31.99
2	GT2-2M Timing Belt - By the Foot	OpenBuilds	OpenBuilds	470-By-the-Foot	470-By-the-Foot	2.49	4.98	2.49	4.98
1	GT2-2M Timing Pulley - 14 Tooth	OpenBuilds	OpenBuilds	960	960	5.29	5.29	5.29	5.29

Figure 27. BOM for One Connect 4 Robot

Appendix C: Power Calculations

12V:

NEMA 17 Motor: 1.6A per coil, 2 coils, *38.4W*

5V:

Servo: 1A, *5W*

3.3V:

MSP: ~100mA, *0.33W*

Photo-Interrupters: ~100mA per sensor, 7 sensors, *2.31W* (extremely conservative)

Total Power Needed: *46.04W*