Smart Chess Board

My Ly Phan, Damani Sinclair, Noel Membribe, and Diego Garcia

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The Smart Chess Board is designed to be a hands-free and voice-activated system that gives the game of chess a twist of livelihood, entertainment, and innovation. To accomplish this, use of voice recognition software as well as stepper motors and an electromagnet were used to control the chess pieces. To further advance the design, extra features were added to make it more fun and aesthetically appealing. This paper goes into detail on the hardware and software used to successfully implement the Smart Chess Board.**

***Index Terms* — Artificial intelligence, electromagnets, engineering, hands-free, speech recognition, voice control.**

I. Introduction

Chess can be a very fun and mind-expanding game, created centuries ago. Though it's beneficial to the mind and played by millions of people around the world, chess isn't enjoyable to everyone. To some, this classic game can be boring, frustrating and just too time consuming. This was where the motivation of the Smart Chess Board came to mind.

The project is similar to any basic chess board but bigger and under the surface of the board, there are electrical and mechanical components that will aid in moving the chess pieces to specific locations specified by the player, making it boxier than the average board game. The use of a laptop is used to run the voice recognition software, as well as for the microphone so the player can speak commands for relocating chess pieces. The base and surroundings of the board was made of plywood, whereas the top surface was partially wood, working as a border, and plexiglass for the playing field. Within the chess board, the electromagnet was attached to the X-Y plotter, that's programmed to slide the specified chess pieces to the locations desired. There were many ideas that were considered in the creation of the Smart Chess Board but because of limitations of time and costs, only a few added features, such as LEDs and an LCD, were included.

II. Structural Design

A very important aspect of the design was the chess board's housing. It holds together the entirety of the project, from electrical to the mechanical components. A great deal of research went into this section because it had to be made to withstand different environments, hold and protect every element of the project, it had to be the right size to avoid extra costs, to be able to fit everything in, and to be as portable as possible.

A. Plywood Housing

The encasement of the chess board was an essential portion of the project for it embodies all of the components of the chess board. Getting the right measurements determined its portability and functionality, and its casing must be strong enough to hold and protect the materials contained inside. It was determined that the chess board enclosure had to be a length x width of 635mm x 635mm and a height of 254mm. Shown in Figure 1 below, the box can be seen.

Fig. 1: Chess Box Housing

B. Chess Board Surface

After much consideration of the major components that would affect surface of the chess board, the plotter's accessibility to go a certain distance in the X and Y direction and chess piece size, the final playing surface of the chess board was organized in a grid format. Each row is labeled with a specific number from 1 to 8 while each column is labeled with a specific letter from A to H. The total Plexiglass has concluding measurements of 635mm x 475mm, the layout of the playing surface with grids can be seen in Figure 2 below.

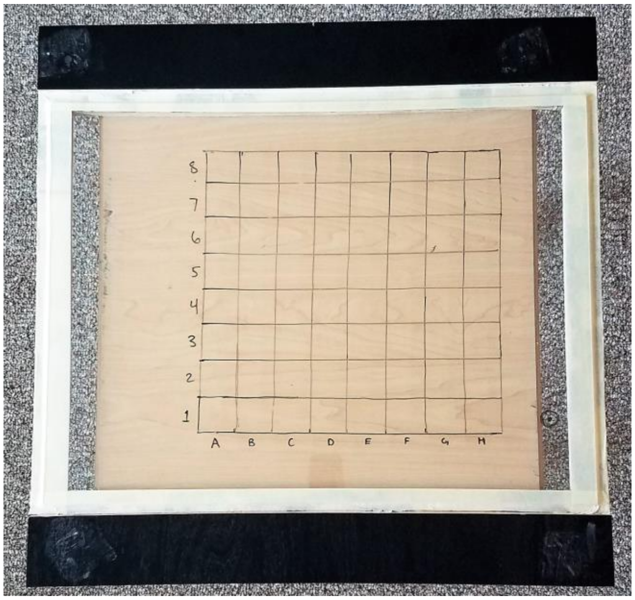

Fig. 2: Chess Board Surface.

III. Integration and Testing

Prototype testing was a crucial step of the design process for saving time and money on product development. For this project, a bottom-up testing approach was chosen. Each component at lower hierarchy was tested individually and then the components that rely upon these components were tested.

All of the software testing in the planning and research phase on this project was to be performed on a team member's personal computer in a Windows environment. The software testing plan consisted of four parts and would be tested separately until each section passes. The software testing plan could not be complete until these four sections pass the required test. The software testing sections included voice capture and recognition, piece movement system, and electromagnet.

The limitations to any design and this design was the hardware. A lot of the hardware testing will be done using the Sunfounder ATMEGA2560 Microcontroller development board. The board was used in conjunction with a breadboard, oscilloscope, power supplies, a multimeter, and other necessary supplies. A signal was to be sent from the microcontroller to a prebuilt circuit on the breadboard just to see if the component actually worked or not, and it helped to understand the best possible way to get the desired output or response. Like the software test, the hardware tests consisted of the less of the voice recognition software, but the piece movement system and the electromagnet was very critical.

A. Voice Capture and Recognition

The voice recognition software required a Voice Schema to be created on BitVoicer to test for accurate interpretation. A sentence anagram is then to be created with a specific data type so the command could be communicated to the microcontroller, done via UART using a USB-to-serial cable to connect to a PC running the application. Testing started simple with a few LEDs turning on and off with code written in Arduino language, which was very similar to C++, and adapted from the build-in Arduino library that came with the BitVoicer application. Each anagram was spoken into the PC's microphone and observed the communication pane on the BitVoicer application to check if the commands were being sent properly. The tester then observed the RBG LED on the breadboard and verified that it turned on and off and changed colors whenever the appropriate commands were spoken.

B. Piece Movement System

The piece movement system, consisting of the XY-Plotter, was tested by writing a test program that could go through all of the possible movements and worst-case scenarios that the XY-Plotter could experience.

The GRemoteFull software package that was included in the XY-Plotter V2.0 Software Manual contained a control program, control software, and G-code examples and was used to manually control the XY-Plotter. The software was downloaded from the software manual and installed on a team member's PC. The XY-Plotter mainboard was connected to the PC using serial communication via UART. During the installation, the proper mainboard and COM port was selected to complete the process.

A set of dummy chess pieces were created out of small blocks of wood, with small magnets attached to their bottoms. These dummy chess pieces were the size of the biggest chess piece, the king, to ensure that the worst-case scenarios were tested. The test performed involved setting up the dummy chess pieces at the same distance away from each other as they would be if they were on the center of a chessboard square on a piece of Plexiglass. This Plexiglass was of similar thickness to the one to be used in the final housing. Then the XY-Plotter was manually controlled through the keyboard shortcuts with the lowest axis increments set up until a piece could be moved in between two other pieces without knocking them over or being obstructed.

The piece movement system hardware test consisted of two components. These components included the XY-Plotter, which was used to transport the pieces along the playing surface of the chess board, and the electromagnet, which was attached to the XY-Plotter to grab and release each chess piece during movement. In each case, the hardware tests must be successfully completed to verify each part before the parts may be implemented into the chess board's design.

C. Electromagnet

The software testing for the electromagnetic was not as extensive as the hardware testing. Since the electromagnet required a supply voltage of 12V, it would not be possible to send a signal straight from the microcontroller to turn on the magnet. With the hardware testing already being completed, the only challenge with the software testing was to see if a signal could be sent from the microcontroller to turn on the magnet. The plan to test the electromagnet was to use the 12V supply as the power source for the magnet, tie the microcontroller to the gate of a MOSFET and use the MOSFET as an on/off switch, which allowed the electromagnet to receive the 12V supply. If the team could accomplish sending a signal from the microcontroller and be able to control the length of time the electromagnet could be one for, then it had passed the software testing phase.

To begin the testing of the electromagnet, the electromagnet to must grab and release each different type of chess piece considered using for this project. Testing began by attaching the electromagnet to the desired chess piece for a period of one second before releasing the piece and continuing to the next one. By completing this task, the electromagnet was proven that it was capable of operating in the way the team intended it to be designed.

Once the previous task was successfully completed by the electromagnet, the electromagnet would be required to grab and releases each different type of chess pieces with a clear plexiglass sheet between the two. The electromagnet would begin by attaching to the desired chess piece for a period of one second before releasing the piece and continuing to the next one. If at any time the electromagnet fails to complete this task, the problem was to be diagnosed, and the task would be restarted.

The electromagnet must then be able to then slide the piece a distance of 300mm while maintaining a strong hold on the chess piece. If at any time the electromagnet fails to complete this task, the problem was to be diagnosed, and the task would be restarted.

IV. SYSTEM COMPONENTS

A tremendous amount of time went into researching, learning, and analyzing different components for this project to be tested for compatibility and functionality. Although Smart Chess Board was a more challenging and demanding choice for a full team that focuses purely on electronics rather than a mixed team with computer or mechanical background, this provided a great learning experience for all of the team members.

A. Microcontroller

There are many types of microcontrollers and microprocessors that could've been used for this design but for the chess board being designed and built to be most efficient, there were some requirements that needed to be met. Some of the more important conditions considered were cost, memory capacity, the amount of General-Purpose Input Output (GPIO) pins, UART and SPI busses, power consumption, and of course, the language support since programming was quite a challenge for a team of all electrical engineering students.

Arduinos are very common boards with a wide range of options that have a microcontroller incorporated within its design. They're affordable, easily accessible, and made for the consumer's convenience in the sense that they're simple to use, customizable, have cross-platform capability, and are open source with a magnitude of software and hardware availability. The ATmega2560, specifically, was decided upon, and its ability to test with the development board made it a very convenient option. The major specifications that aided in the selection of this microcontroller was its CPU speed of 16MHz, program memory size of 256KB, RAM size of 8KB, 86 general input/output pins, interface of I2C, SPI, and USART, clock speed of 16MHz, operating voltage of 5V, and at a price of approximately $14.

B. Voice Recognition Software

A major focus of this project was to create a product which allows users to play a game of chess using only voice commands to control their pieces. This was implemented through the use of a voice recognition software which converts the user's voice commands into a set of text instructions which can be inputted to the piece movement system to produce the appropriate movements. Research was done on the different open-source voice recognition software packages available until one was chosen that appeared easier to use and could've been implemented into our project.

Voice or Speech recognition can be a computer software program or a hardware device with the ability to decode human voice. Voice recognition omits having to use a keyboard or press any buttons in general. This portion was executed on a computer with an automatic speech recognition (ASR) program. Depending on the ASR, it could require the user to train it in order to recognize the users' voice. Because the chess board required commands from multiple people to be recognized, was going to have to choose a speaker independent system. The speaker independent system would be able to recognize most users' voices with no training.

The voice recognition program that the team had decided to go with was BitVoicer. It's a speech recognition application that enables simple low processing power devices to become voice operated. It analyzes audio streams to identify sentences in these steams and then sends commands to the microcontroller connected to it. It supports many different languages and has the ability to store numerous commands and sentences. It also uses a USB/serial or TCP/IP communication interface. The benefit about using this software was that there are many projects that uses features similar to the smart chess board posted on the website, which can be referred to at any time.

C. Electromagnet

One of the main features of this design was to not have to physically move the chess pieces, and to successfully execute that implementation, magnets play a crucial role. The magnet that was attached to the piece movement system needed to be able to "pick up" any of the chess pieces by attracting it, via its magnetic field, through the surface of the plexiglass chessboard. It needed to be able to hold onto the pieces after grabbing them and slide them across the board smoothly. The piece movement system could only support movement in two directions, the X and Y plane and cannot go out of the playing field. Therefore, the magnet needed to be able to be switched on and off easily so that a piece can be "dropped" or let go of.

Selecting the most compatible type of electromagnet was a bit complicated because a great deal of variables needed to be considered. The electromagnet needed to use the least power as possible while maximizing the holding strength. It had to be able to work with DC voltages and it should not be too big, otherwise it would attract all the other pieces around the targeted piece. The electromagnet the team chose was the Uxcell 12V 180N electric lifting magnet, shown in Figure 3.

Figure 3: Electromagnet.

Because the electromagnet needed 12V in order to operate, it was required to be powered straight from the wall adapter and be turned on and off using the signals sent from the microcontroller. The microcontroller was to be tied to the gate of a MOSFET and it would act as a switch allowing current to flow through the electromagnet. The electromagnet and the XY-plotter would then have to work in unison.

D. Piece Movement System

For this voice-activated chess board design, the players are to verbally make commands by communicating with the voice software. The player is to state the desired piece to activate, specifically, where the piece was located on the chess grid, and then declare where the destination is to be. After the speech is recognized by the software, the selected chess piece would then be moved hands-free around the board to the operator's desired location. In order for this to be possible, a mechanism had to be implemented into the design to attach and move each chess piece without coming into contact or disturbing the other stationary pieces on the playing surface.

With the playing area of the chess board being a flat surface with both an X and Y plane, the electromagnet was required to move freely in both the X and Y directions mimicking the path and motion of chess pieces during normal play. To accomplish this, it was decided that an XY-Plotter would be implemented inside the chess board's structure to move the electromagnet to the desired chess piece and transport said chess piece through the path of the commanded move to its finishing location. An XY-Plotter has free range of motion in both the X and Y directions and can travel freely and precisely within the designated area of the plotter's specification. For this hands-free chess board design, an XY-Plotter could be built, or a kit could be bought, modified and implemented under the board to follow the given paths commanded and move pieces using an attached electromagnet with the same precision seen in other applications of this device. The XY-Plotter built and used could be seen in Figure 4.

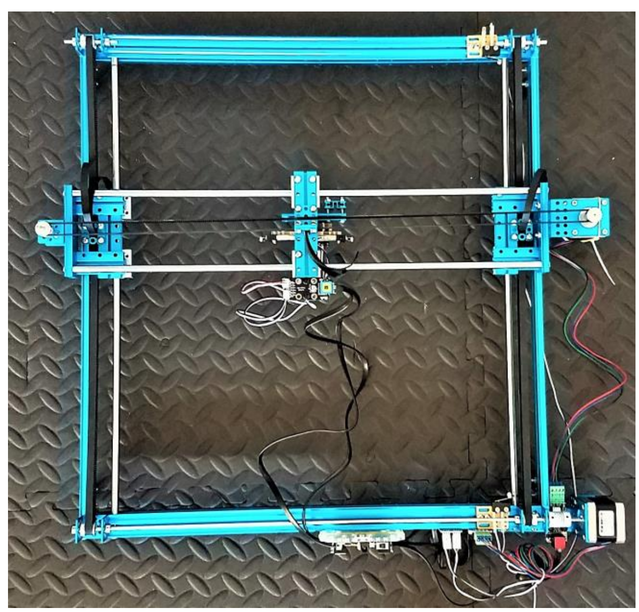

Fig. 4: XY-Plotter.

E. PCB Design

EAGLE design software was utilized to draw the schematic and the PCB. Figure 5 below demonstrates and shows a schematic of all the components that were connected to the ATmega 2560. Starting from the top left, there was a 12V source coming from the wall outlet into the LM2596 step-down adjustable DC-DC switching buck converter. That same 12V source power also energized the electromagnet and the XY-plotter. Focused back to the regulator, it drops the 12V source down to 5V in order to provide a suitable power supply for the microcontroller to work. From the microcontroller, several peripherals were either supplied a source of power, sent PWM signals, or sent communication signals.

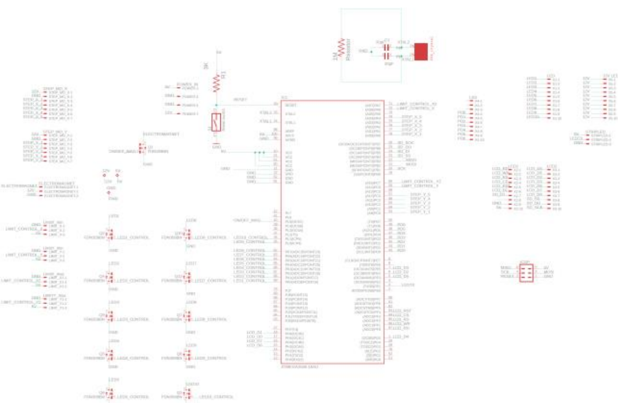

Figure 6: Microcontroller Schematic.

From the research done, a Rigid board would be the most practical choice to use for the chess board. It is the most common type of PCB out there and the most practical considering the application. The team ended up creating two iteration of the PCBs. The first design was manufactured by JLC PCB in China. This was a great choice because it ended up being a fraction of the price as compared to other manufacturers. The downfall with this was the estimated time to receive the board.

The first PCB that the team ordered had numerous errors, it wasn't critical but just caused an inconvenience when trying to work with the chip. The wrong size header pins that were placed on the board caused the team to have to attach a 22-gauge wire to each pin header in order to use them. The last reason was that there weren't enough GPIO pins available to control the necessary peripherals; this was caused by a change in the design. The first PCB design can be seen in Figure 6.

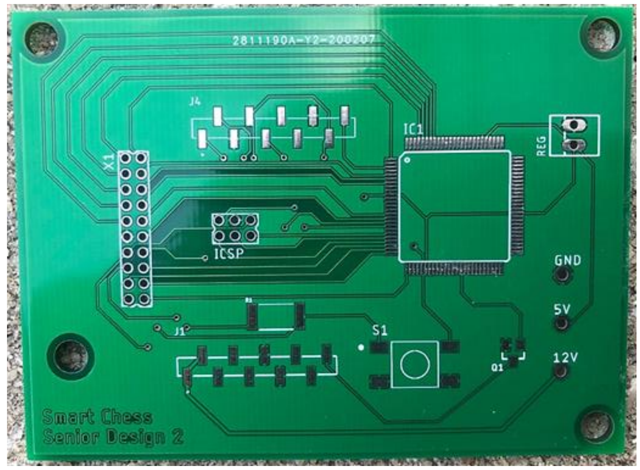

Fig. 5: First PCB design.

When first trying to program the microcontroller on the PCB, the team decided to go the cheaper route and use the development board as an in-service-programmer ISP. This method was definitely a bad choice because we lost countless hours trying different ways to program it. There were instructions to attach a capacitor in between reset and ground on the development board, some instructions even said to use a resistor. Code was also altered in order to target the correct pins and it still didn't work.

The team decided to use the method of using the Atmel-ice programmer. It was the programmer that was needed to specifically, program AVR chips and it was used as a last resort because it was costly. After trying to program the chip with the Atmel-ice, the team was finally able to upload code. Testing was done by an LED at a random GPIO pin and seeing if the team was able to make it blink the detaching programmer and observe the continuous blinking.

After finalizing that we were able to upload code and verified that the Schematic wasn't the problem, the team decided to place the order for the final PCB design. There were many changes that were made to the final PCB board but a few of the main changes were; placing the correct size pin headers, using more GPIO pins, grouping and placing parts correctly, and adding more components. When the board finally arrived, the team decided to go ahead and test it by programming it with the Atmel-ice and seeing if an LED could be flashed. It worked without a problem. In the Figure 7 below, the final PCB design can be seen.

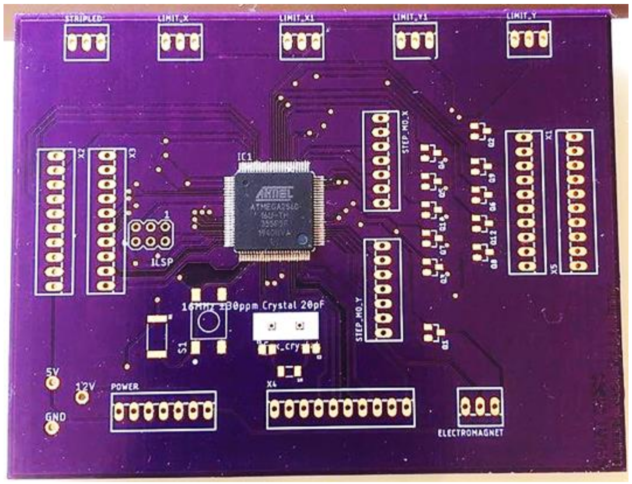

Figure 7: Final PCB design.

F. PCB Programmer

The original plan for the team to program the PCB board was to use a development board as an In-System Programmer (ISP) in order to flash code onto the PCB. This method was highly ineffective and just made things more difficult. It was more difficult because it took a lot of time to individually find each programming pin, the wire could easily be mixed up, and the instructions and tutorials were not correct. Weeks of valuable time was spent trying to flash code to the microcontroller using this method and it never worked. The team then switched to using the Atmel-Ice programmer, shown in Figure 8, which is a device meant to program AVR chips. After switching to this method, flashing code to the chip became relevantly easy and now possible. The programmer came with one wire that connected to all six programming pins at once which eliminated the individual plugging of wires that would normally allow room for error. The Atmel-Ice programmer also used a program called AVR studio that was user friendly and allowed easy configuration of the clock and different fuses.

Fig. 8: Atmel-Ice Programmer.

F. Power System

Since standard wall outlets was used to fully power the device, the maximum power that could be used was 1440W. This was calculated based on a 120V, 15A outlet under a continuous load. Using any more power than that would cause the breaker associated with that particular outlet to trip and shut off the power to that outlet. Regardless, power consumption should be kept as low as possible for a host of other reasons. The less power used, the less heat would be generated by the device, which extends its life and prevents fire hazards from overheating electronics. Minimizing power consumptions leaves open the possibility of a fully battery-powered device. The intent was to first create a device which could be powered by a standard 120V wall outlet, however, if it was possible, then battery capabilities could be added as an optional feature.

The AC adapter from the wall would convert the AC power to DC power using a transformer, rectifier, and other electronic components that cleans the signal up in order to produce a constant DC signal. At this point everything else would be operated using DC. The team used a 12V 8A DC wall adapter as the power source. It was used to power every component associated with the chess board, the XY-plotter, the electromagnet, LEDs, etc. The 12V supply was connected to a buck switching DC-DC voltage regulator in order drop the voltage from 12V to 5V, this supplies the microcontroller and all the other necessary devices.

G. LEDs

Since the lighting option for this design was more for entertainment and cosmetics rather than a learning tool, using an RGB LED strip would've be the most accommodating choice. These LED strips have the ability to turn on, off, or flash a wide range of colors and options, depending on what they are programmed to do. The plan was to align the LED strip along the edges of the top surface of the chess board and have them display different colors depending on the circumstances. When the game begins, the LEDs were to continuously light up from one end to the other. Each player would be assigned a different color to know whose turn it is. When a chess piece was to be killed off the board, it would flash red a few times, when a piece was to be promoted, i.e. when a pawn reaches the opposite side of the board, it would flash blue, and when the game ends, the color of the winner will flash.

There were many choices of LEDs to choose from, considering brand and specifications, but as a team, NeoPixel by Adafruit were the ones that stood out the most. They come in all different sizes and shapes, through-hole single LEDs, in a strip, in a ring, and as well as in a matrix. Its option of being purchased with an integrated driver also made it a lot more convenient and simpler to add to the chess board.

The Adafruit NeoPixel Digital RGB LED Strip was affordable and flexible. Having a rubbery casing and bendable PCB strip, along with the width of 12.5mm with the casing and 10mm without, and 4mm thickness with the casing, makes it very accommodating to avoid interference with any of the other components. Not to mention the benefit of it not taking up much space, promoting the portability aspect of the chess board. The maximum amount of wattage required for these strips are 9.5W, 2 amperes at 5 volts per meter. When illuminating in white, the maximum was required, whereas the other colors only require a third to a half of that maximum current. The chip integrated in the strip used a single input pin and a single output pin, controllable by a microcontroller with a 100nS highly repeatable timing precision, making it very cooperative.

The development board and microchip the team decided to test on was the ATmega 2560-16AU board, and with this LED, they're compatible and have the capability and support to easily be controlled by programming them. It has the option to individually be controlled, with the pulse width modulation installed in each and every LED, to whatever the desired color. The flexible material it's made of is also weatherproof and can be cut to an appropriate length to fit inside the board. The LED strip used can be seen in Figure 9.

Fig. 9: LEDs.

H. Communication Features

To provide an interface between the system and players to communicate moves hands-free, it was essential to have some sort of connection to instruct and direct. A great deal of research went into this section of the project and identifying a capable product at a reasonable price was a bit difficult. So to make it easier for the team, it was discovered and decided that the main means of communication from the player and the voice software would be a headset, or a simple pair of earbuds that had a microphone, for the user so the system can clearly understand what was being said. The team soon realized not long after that because the software was running off of a laptop, it was also possible to use the default microphone installed on the laptop, as long as the surrounding isn't so noisy that it'll cause a disruption or false readings of what's being said.

While researching the different products for this section of the project, a few things had to be kept in mind, compatibility to the microcontroller, cost, and size. The main purpose of the LCD screen was to visually see what was supposed to be happening on the playing field, as well as what was being communicated by the players and the instructions by the voice software. The team found a relatively inexpensive LCD screen that was compatible with the Arduino.

After testing, the team found that the Elegoo UNO R3 2.8-inch TFT Touchscreen was the best option. The seller provided plenty of information, resources, and plenty of sample codes to get started. Some other advantages of this LCD included the support of the Mega 2560, had a 240 x 320-pixel resolution, a PCB module display size of 78.22mm x 52.7mm, an 8-bit parallel interface, and compatibility with 5V or 3V MCU with 5V-3.3V change-over circuit. Figure 10 below shows the LCD display embedded within the chess board.

Figure 10: LCD display in the Chess Board.

V. Conclusion

The design the team were aiming towards was a hands-free and voice activated chess board that included extra features for a more entertaining and aesthetically pleasing product. We certainly learned an ample amount of information on the procedure to design and implement a plan creating the Smart Chess Board. Senior Design I required an abundant amount of the research and product selection process and Senior Design II revealed the need of hard work and dedication to successfully succeed in the project. It will serve as a strong backbone to sufficiently approach future projects in the work field and provide an impressive background.

BIOGRAPHIES

My Ly Phan is a senior attending the University of Central Florida and will be graduating Spring of 2020 with a Bachelor's degree in Electrical Engineering. She is working as a test engineer at an advanced antenna technology company and is hoping to expand her knowledge with more future work experience.

Damani Sinclair is a graduating senior at the University of Central Florida in Electrical Engineering. He is a product engineer and hopes to continue that path into product design.

Noel Membribe is graduating from the University of Central Florida with a bachelor's in Electrical Engineering. He works as a software developer in the space industry and looks forward to continuing that path after graduation.

Diego Garcia is a graduating senior in the electrical engineering program at the University of Central Florida. He plans to continue his career at the Kennedy Space Center in Cape Canaveral, Florida working for Jacobs Technology.

ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance and support of Idoku Abuh and Dr. Samuel Richie.