ADC/DAC Swordle

A Technical Report submitted to the Department of Electrical Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Raymond Costa

Fall 2022

Technical Project Team Members

Raymond Costa

Lillian Price

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

Harry Powell, Department of Electrical Engineering

ADC/DAC Swordle

Lilian Price, Raymond Costa

May 12, 2023

Capstone Design ECE 4440 / ECE4991

Signatures

Lilian Price

Raymond Costa

Statement of work:

Lilian Price

My primary role in this project was the design of the hardware system. This included tasks such as choosing components that were within our power budget, designing the schematics, designing any custom footprints, and creating all the wiring routes for the entire system. In terms of the schematic, I was responsible for creating any custom footprints that did not have models already available with the component part online. Additionally, I accounted for our testing procedure in the design by adding color-coded test point pins on our primary supply lines for testing after the board was fabricated. At the same time as these tasks, I was responsible for creating the necessary documents to go alongside the PCB fabrication which included the parts order list and the bill of materials (BOM).

Upon the arrival of the fabricated board, I was responsible for drawing up the documents for WWW Electronics Inc. (3W) in order to have select components soldered onto the board. After the board was returned, I was responsible for using a digital multimeter (DMM) to test each power supply route through the various test points and make any needed adjustments, which in this case required some custom wiring. The overall power supply was tested numerous times before attaching each display component to the board, such as the display itself and the keypad, where the power supply was tested again to ensure that the supplies would not damage any of the components. Overall, I was responsible for the entire hardware aspect of the system, with received assistance from Raymond, in order to power the various components that are essential for the proper function of the game without causing any damage to the game components.

Raymond Costa

My primary contributions to the project came during software development and testing. I led development of all three major external communication components of the project (serial flash memory, keypad, and display) as well as the internal game logic within the MSP432. This included developing the game logic, determining what subsystems needed to be developed, and then developing and testing each subsystem. As far as my secondary contributions I aided in developing hardware components by reading data sheets and exploring component options, as well as developing the power supplies component of our PCB.

From a research perspective I primarily aided in researching prior work and potential sources of copyright or patent infringement. Additionally, I helped determine potential standards that needed to be met in terms of electrical supply, programming standards, as well as safety concerns for household toys.

Table of Contents

Contents

Table of Figures

Table of Tables

Abstract

Swordle is physical game console that allows two opponents to play a word guessing game against one another that loosely resembles the New York Times online game Wordle [1]. The first player enters a word that is between 2-5 letters onto the display screen, and the second player is prompted to guess potential words, with the number of guesses corresponding to the number of letters in the word (i.e. a three letter word means the player is given 3 guesses). After each guess the player is given prompts to aid in any upcoming guesses which are the letter appearing green, meaning it is the correct letter in the correct placement, or yellow, meaning it is the correct letter but in the wrong placement. The game console consists of a power subsystem that communicates to a keypad and display via an MSP432 microcontroller. The software aspect of the system uses a laptop to initially load in a dictionary, and then is a stand-alone console that uses the communication between the keypad and display to conduct the game, with the only connection being a standard 120 V AC outlet plug-in.

Background

The occurrence of the COVID-19 pandemic almost instantaneously forced millions of people into prolonged isolation. Apart from the struggles that came alongside evading a potentially dangerous disease, living in prolonged isolation can be almost equally harmful towards one's mental and physical health. A recent study, published within the last year, found that there had been a significant increase in suicidal behavior, which is a direct result from the pandemic and the extreme isolation that many worldwide had to experience [2]. As we now slowly return to normalcy, activities that encourage socialization are beginning to pull people away from the previous year of isolation.

The purpose of this project is to create a game console where players can engage in a word guessing game against one another. The game itself, which is inspired by the NY Times game Wordle, serves as a way for people to interact with each other in a lighthearted manner. In order to play the game, the user is prompted with a game mode, which is selected between player vs. player, player vs. computer, and computer vs. player, and a number of letters which ranges from 2-5 [1]. Depending on the chosen game mode, the first player will enter a word that the second player, or computer, has to guess, with certain clues given at the end of each guess to guide future attempts. After inputting a guess, the letters will either remain their original color or turn yellow to indicate the correct letter but in the wrong spot, or green, which indicates the correct letter is in the correct spot. In each round, the number of guesses is equivalent to the number of letters in the word. What differentiates our game from other similar web-based games, such as Wordle, is two-fold [1]. The first being that you can select the amount of letters in the word you are guessing, which makes the game accessible for a larger age range. The second difference is that you can also select the word which is being guessed, whether that is by the computer or another person, allowing for a multiplayer mode and greater functionality.

Our design was chosen with the explicit intention of creating a way to bring people together, especially across a large age range. The idea behind creating a game console was to not only pull people away from our typical reliance on phones, but also to create a sense of nostalgia from using an older style of game console.

The construction of our game console design pulls information from a variety of courses. In terms of hardware design, the ECE Fundamentals courses provides the majority of the background information that is needed to construct the power subsystem. Additionally, the software aspect of the project relies heavily on information taught in Embedded Computer Systems (ECE 3430), which allows our game display to function via the MSP432 microcontroller. Alongside this, basic programming knowledge from Introduction to Programming (CS 1110) and other core CS courses such as Software Development Methods (CS 2110) and Data Representation (CS 2150) provides background knowledge in coding in C/C++, which is necessary for designing the software communication between components and game rules.

Physical Constraints

Design Constraints

The primary design constraint was in relation to the CPU we were working with, which is the MSP432. Our team chose the MSP432 due to the 20 I/O interface pins on the header board in addition to multiple types of interfacing methods such as SPI and I2C. Additionally, the MSP432 already had some built-in storage to test initial libraries.

Cost Constraints

The primary economic constraint for this project was to stay below $500 for the design and fabrication of the entire system. This budget was used for not only the fabrication of the PCB itself, but also for all the components that were used to populate the PCB, the game components such as the keypad and display, and any materials for the enclosure. Any other components already available to our group through the University, or testing equipment such as a digital multimeter, was not factored into our budget which kept our overall cost within the $500 range.

Tools Employed

In order to create a finalized product, an extensive list of tools were used in order to design, fabricate, and test our game system. The breakdown for the tools used in each category between hardware and software is expanded on below.

Hardware:

To create the original schematic, which included the wiring and physical PCB layout, KiCad 7.0 was used [3]. Within the KiCad software, we were able to import and create customized footprints for each of the component parts, and place all the wiring between the footprints on the schematic. The KiCad software was new for both team members, so it required a lot of independent research in order to find methods to create custom components on our board. After the PCB layout was complete, the FreeDFM online service from Advanced Circuits [4] was used to verify that the PCB board did not contain any manufacturing errors and was in fact ready to be fabricated. The last tool in the design process was 3W Electronics, who populated the board with our components [5]. In terms of testing, a generic digital multimeter was used to verify the power supply tracks throughout the entire circuit using the test points.

Software:

To interface between subsystems and the MSP432 microcontroller, the Texas Instruments' integrated development environment (IDE), Code Composer Studio (CCS) was used [6]. In order to verify a few of the subsystems the MSP430 library in CCS was used which applies to any microcontroller within the MSP430FR2XX_4XX family, and uses preprogrammed functions to interface to the MSP432 [7]. Both members of the team were familiar with CCS since it uses the same IDE and microcontroller used in Introduction to Embedded Computer Systems.

Societal Impact Constraints

Environmental Impact

The system itself relies on the power from a standard 120 V outlet supply. This creates a long lifespan for the game, given that there are no components that need to be regularly replaced, like a battery source. Additionally, given the enclosure type, the longevity of the internal system is ensured since the embedded system and screen are protected from outside damage. One of the primary environmental concerns arises from the disposal of the device. Although this concern is limited due to the fact that there are no components being regularly thrown away or replaced, it is important to consider the disposal methods for the device. Not only do the plastic components of the device release microplastics into the environment if left to break down in a landfill, but those pollutants can lead to the quick spread of harmful volatile compounds that rapidly spread through the air due to their small size [8]. Alongside this, electronic components typically contain toxic chemicals and metals that can prove to be harmful if disposed of in large quantities. Although the disposal of electronics is not federally regulated, the United States Environmental Protection Agency (EPA) provides guidelines for the proper way to recycle electronic components in each state [9]. These risks are mitigated by ensuring that the components are properly disposed of after the lifetime of the device has expired.

Sustainability

The overall game console system creates a sustainable design given that it is built for long term use without replaceable parts. The design is powered entirely by a standard wall outlet, as opposed to using batteries which would need to be replaced regularly. Additionally, the enclosure itself is not made from harmful plastics that could potentially release harmful chemicals into the environment [8].

Health and Safety

Given that the design is marketed for all age groups, the primary concern is ensuring that the system is resistant to potentially discharging any electricity from the embedded system or LED screen to the user. This concern is taken into consideration by keeping the embedded system and screen enclosed in a physical structure, with only the plug-in opening being accessible to the outside of the box. Additionally, the system is resistant to outside damage, so the user should not experience any electrical shock from water or dirt coming into contact with the enclosure. An additional concern arises from the game being used by children, so the device design is in accordance with the standards outlined by the Code of Federal Regulations for the manufacturing of electronic devices that are marketed to children, such as proper labeling and directions in addition to displaying warnings on the device that are easily understood by children, which is outlined in Title 16, Chapter 2, Subchapter C, Part 1505 of the Code of Federal Regulations [10].

Ethical, Social, and Economic Concerns

The intended purpose of this project is to encourage collaboration between individuals of all ages. However, there are a few ethical concerns that arise in relation to fair use. In its current form, the game is only available in English, so those who do not natively speak English cannot fully participate in playing the game. Furthermore, any individuals who are blind or visually impaired and suffer from conditions like colorblindness, cannot participate in the game given its heavy reliance on the use of colors. In order to make Swordle accessible to a wide range of socioeconomic groups, it is important for the cost of the game console itself to remain as low as possible. However, given the nature of the game requiring multiple players, it is unnecessary for individuals to purchase their own game console, so the cost per individual decreases if the game console is shared as it is intended.

In order to increase the accessibility of the game, the audio capabilities of the display could be used in order to add verbal cues for those who are visually impaired. Additionally, the cost of the overall game system itself would decrease if the parts were purchased in large quantities and a cheaper LED screen was found as an alternative to bring the price down further.

External Considerations

External Standards

The relevant external considerations for the Swordle design apply to not only the hardware and software system, but also the physical enclosure itself. A comprehensive list of the external considerations taken into account for the design of the game console are listed below.

1. *Barr C Standard* – The outlined coding standard provided by the Barr C group serve as a basis for software development in order to ensure that the maintenance and code align with industry conventions [11]. These standards range from syntax rules such as spacing and formatting functions to properly laying out your code.
2. *PCB Design Standards* – The general design parameters for fabricated PCB boards is outlined by IPC standards. Each of the standards serves to monitor certain aspects of the board such as wire thickness, spacing between components, etc. [12].
3. *Surface Mounted Components (SMD)* – All components that are surface mounted must abide by the guidelines given in the Surface Mount Technology (SMT) package. These specifications are explicitly outlined in JEDEC, which specifies size of any surface mounted components and were considered during the layout process [13].
4. NEMA Enclosure Standards – One of the primary physical design components to the Swordle is the waterproof and damage proof casing, which are required given that the game is classified as being portable, so it will likely be used both indoors and outdoors. With this design goal in mind, the prototyping and manufacturing of the game board must fall under the classification of a NEMA 3 enclosure, which protects the system from foreign objects such as water and dirt causing internal damage [14].
5. Shock Hazard, Occupational Safety and Health Administration (OSHA) – In order to be manufactured as a game available to all ages, especially including children, OSHA outlines key standards associated with the wiring and enclosure in order to eliminate the potential for shock [15].

6. Intended use for children – For the specific intended use for children, there are compliances required in terms of electrically operated toys that fall within the standards outlined by the Code of Federal Regulations in addition to safety protocols given by the National Institute of Standards and Technology [16][17].

Intellectual Property Issues

This design project does not, at this stage in the development process, have the ability to receive a patent. Three patents that produce a similar product to Swordle in one aspect or another are described below.

One previous patent, called "Word Game" is described as a game where "one or a plurality of players try to correctly identify an unknown word with the minimum number of opportunities" and the game lets players use feedback on previous guesses to guide their subsequent guesses. Although there are similarities between this previous patent and our current design, the entire physical system needed to participate in the game is vastly different, as ours uses an embedded system with a screen display and the previous patent utilizes a mechanical system [18].

Another previous patent, referred to as "Apparatus for Word Guessing Board game" is similar in rules to our current design. Players are able to guess an unknown word, expect in this case, their guesses are unlimited. In terms of the enclosure itself, the game consists of a deck of cards with an apparatus that accompanies it for the players to manually make their guesses. Although loosely similar, this previous patent resembles out design closely in terms of game rules and having a physical apparatus to make guesses, however the design lacks and form of an electrical component to the game [19].

Another previous patent, referred to as "Word Game Device" resembles our design based on their added feature that "successive comparisons are made with different known words to determine the unknown word." In a similar fashion to our design, the game is centered around players guessing an unknown word where their previous guesses are used to aid their subsequent guesses. This game also mirrors Swordle by having a variety of word lengths, much like our option of having 2-5 letters. However, in this case, while also lacking a similar game console design, players are only given the option to play against the cards already within the deck, so players are not permitted to play directly against one another [20].

At this stage in the development process, our design closely resembles a variety of games that already exist, and against most previous patents, our game console is the differentiating factor. In order to truly determine if our game could get a patent, more research would need to be done in terms of the game console design specifically.

Detailed Technical Description of Project

The goal of the project was to design a game console that would allow two players to engage in a word guessing game. The overall block diagram can be seen in Figure 1 below, which provides a general overview of the components of the project.

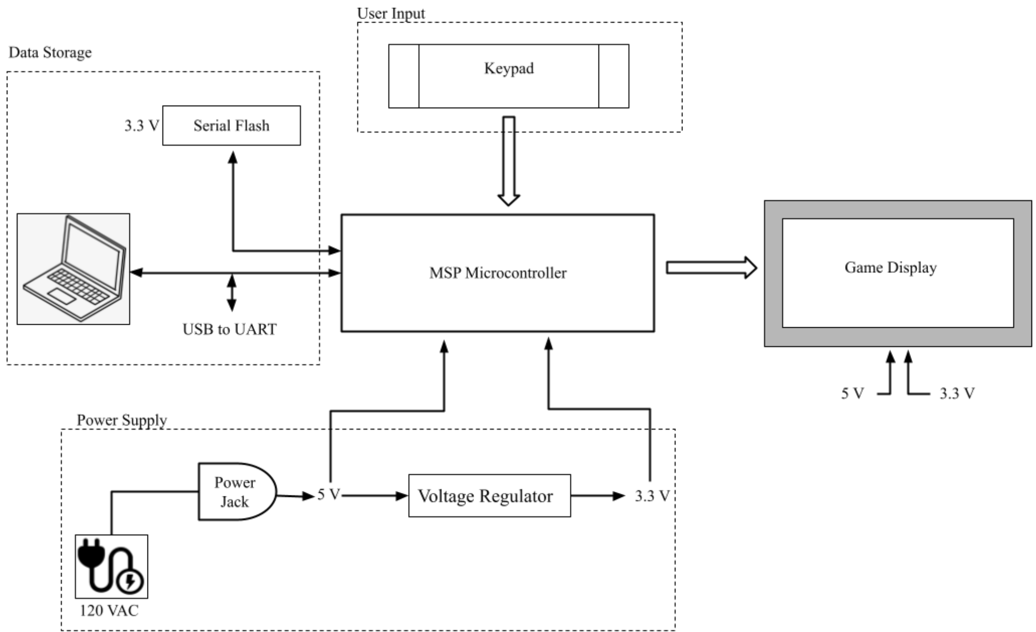

Figure 1. System Block Diagram

Figure 1 displays the overall system block diagram for the Swordle game console. The entire game console is run by the power supply which connects the console to a standard wall outlet in order to supply the subsystems with either a 5 V or 3.3 V supply, which is differentiated through a voltage regulation system. Each subsystem interfaces to the MSP432, which controls the communication between the keypad and game display, in addition to subsystems such as the serial flash which stores the dictionary and USB to UART COM port. The MSP432 first communicates with the system through the display, where the player is prompted to select a game mode and insert a word via the keys on the keypad. The MSP432 then cross-references guesses against the loaded dictionary in the serial flash in order to verify the validity of the guessed words.

The overall system can be broken down into the hardware components used and software components. The hardware system responsible for powering the entire board can be broken down into its respective components below:

1. Voltage Regulation System
2. USB-UART COM Port
3. Serial Flash
4. Keypad
5. Display
6. MSP432 Microcontroller

In terms of the software components of the system, the overall design can be broken down into its respective subsystems:

1. Keypad
2. Serial Flash
3. Display
4. Game Logic

Hardware

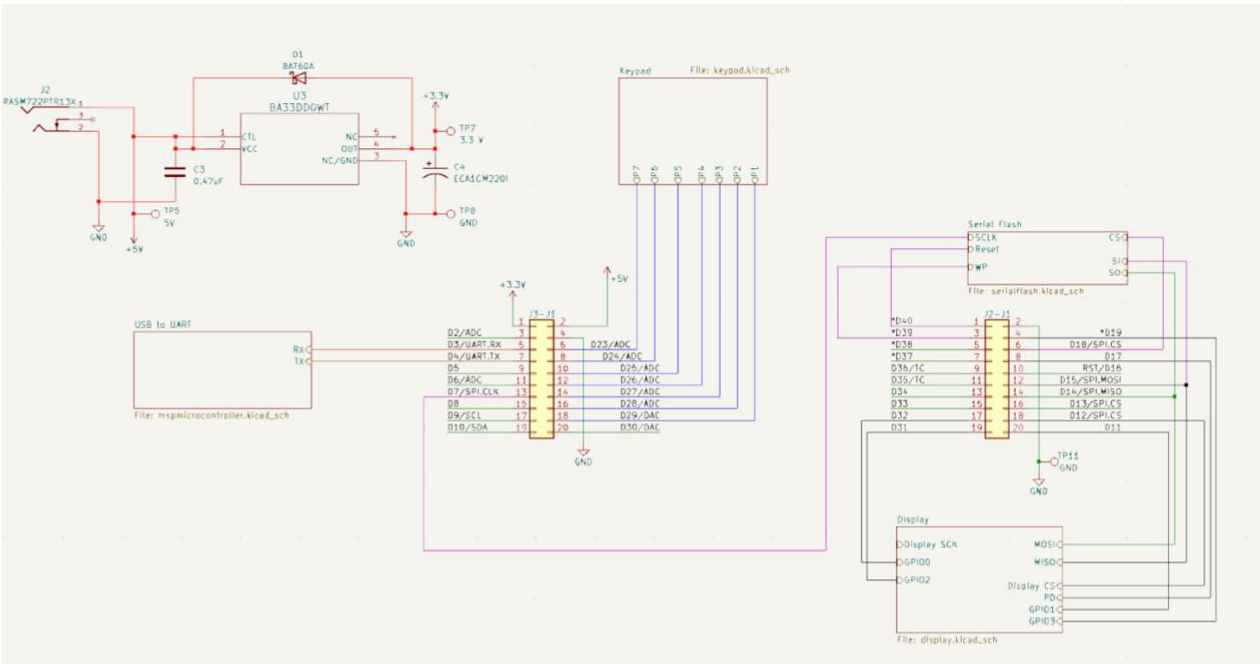

Figure 2. Power Supply Schematic Overview

The hardware system, which can be seen above in Figure 2, consists of a PCB that is designed to be mounted onto the 20 I/O interface pins on the MSP432. The overall system is composed of a voltage regulation system, a USB to UART COM Port connection, a keypad interface, serial flash connection, and display screen connection. The voltage regulation system is responsible for creating a 3.3 V output voltage that is used to power the serial flash and game display supply while also having access to the 5 V input supply to power the display backlight. The entire power supply system is generated from a power jack that plugs into a standard 120 V AC outlet. The display screen interacts with the keypad by manipulating the display based on the selected input from the keypad. The inputted words are then cross checked against the stored dictionary in the serial flash in order to verify that each guess is legitimate. The USB to UART COM port provides the ability to upload a dictionary via the MSP432 that is then stored in the serial flash. The overall schematic shown in Figure 2 was constructed in KiCad, and each subsystem is discussed in further detail in the coming sections.

Voltage Regulation System

The Swordle game console is powered through the wall outlet that connects to the power jack on the PCB (displayed as component J2). The voltage regulation system can be seen below in Figure 3.

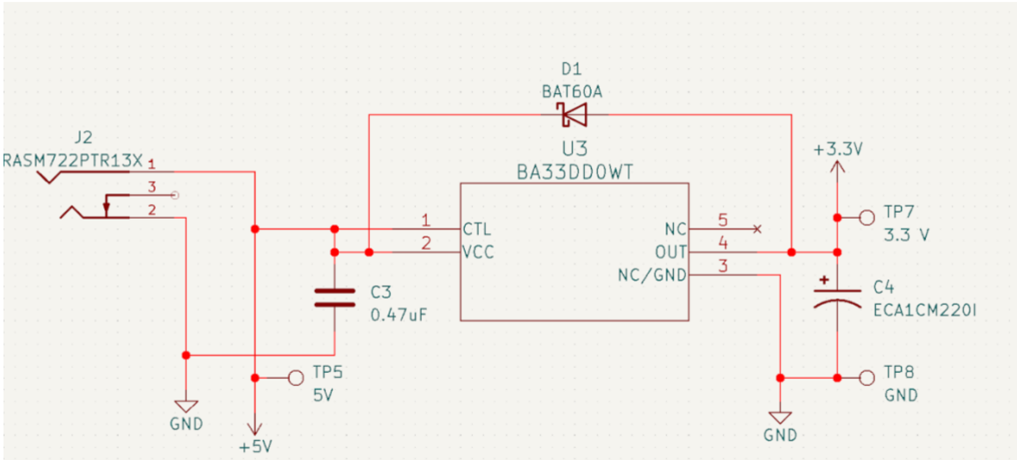

Figure 3. Voltage Regulation Subsystem Schematic

In terms of functionality, the voltage regulator intakes a 5V supply, which is verified through test point 5 (shown as TP5 on the schematic), and creates a 3.3V output supply, which is verified through test point 7 (TP7).

The total current budget for the BA33DDOWT voltage regulator is 2A. The total current drawn from the entire system itself is broken down below in Table 1:

Table 1. Current Breakdown for System

Component:	Current:
USB-UART COM Port	100 mA
MSP432 Microcontroller	100 mA
Serial Flash	22 mA (max)
Display	760 mA (max)
Total Current Consumption:	982 mA

The maximum current consumption for the entire system amounts to 982mA, which is well within the 2A range provided by the voltage regulator.

Serial Flash

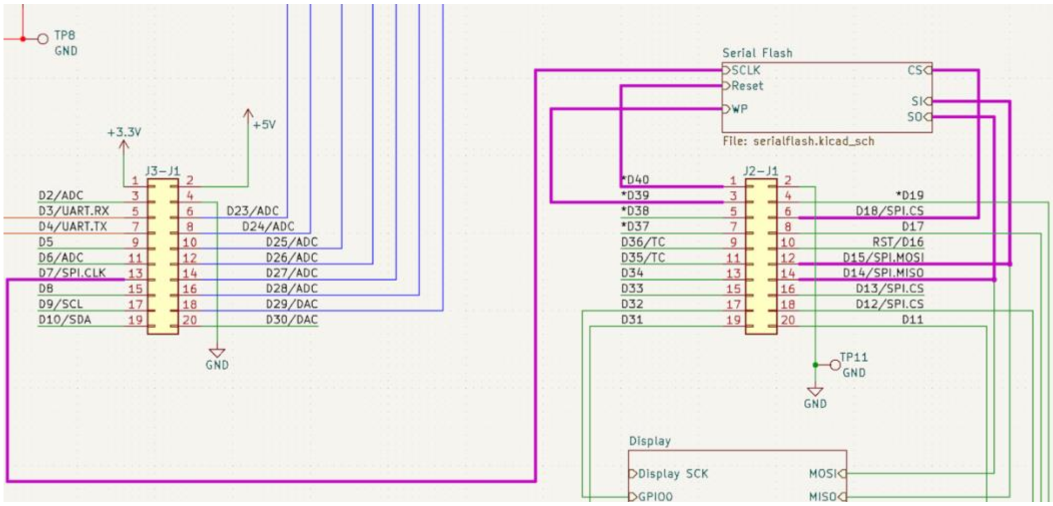

Figure 4. Serial Flash Hierarchical Schematic View

The overall hierarchical structure of the AT45DQ321-SHF-T serial flash can be seen above in Figure 4. A view of the subsystem itself can be seen below in Figure 5.

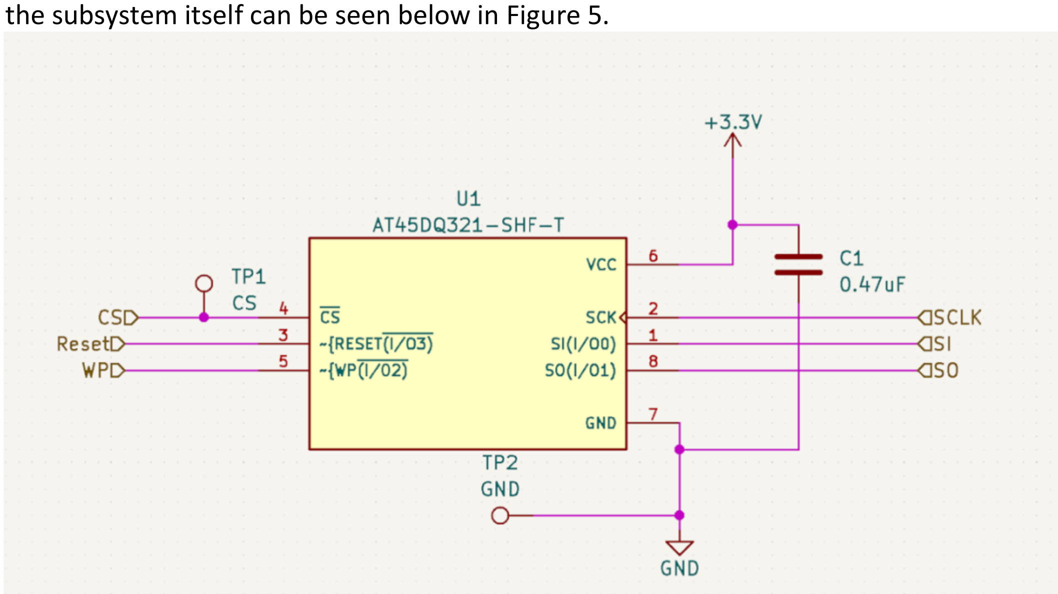

Figure 5. Serial Flash Subsystem Schematic

The serial flash system consists entirely of the connections between the chip itself and the MSP432, which the relevant pin labels shown on the schematic. The two primary electrical test points for the serial flash system are TP1, which is on the chip select pin, and TP2, which is placed on the ground supply. The serial flash relies on the 3.3V supply from the regulator in order to shift the desired dictionary onto the serial flash, via the MSP432, to be stored.

Table 2. Serial Flash Pin Connections

Connection	Type	Function
CS	Input	Chip Select
Reset	Input/Output	Reset
WP	Input/Output	Write Protect
VCC	Power	Supply Voltage
GND	Ground	Ground
SCLK	Input	Serial Clock
SI	Input/Output	Serial Input
SO	Input/Output	Serial Output

A detailed list of the relevant connections to the serial flash is visible above in Table 2 with their corresponding functions

Keypad

The hierarchical view of the keypad is shown below in Figure 6, and the subsystem view itself can be seen in Figure 7.

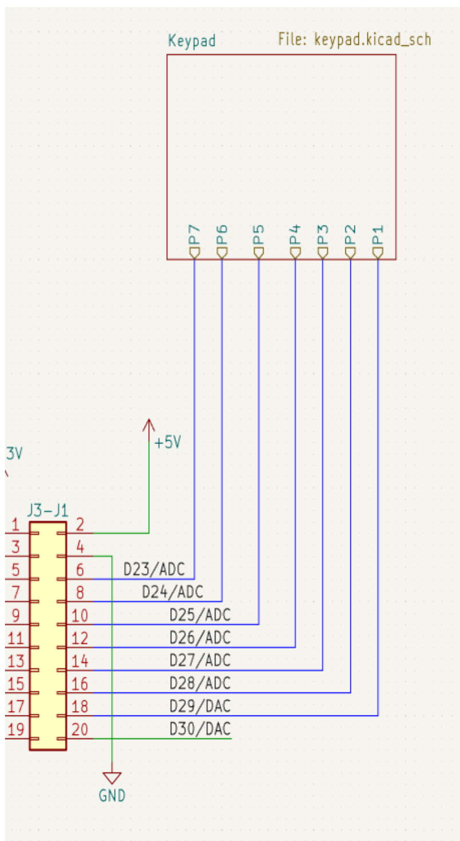

Figure 6. Keypad Hierarchical Schematic View

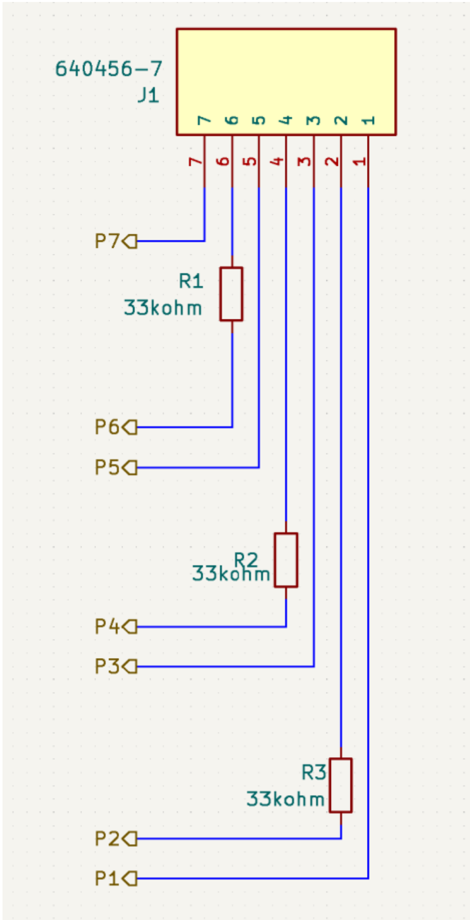

Figure 7. Keypad Subsystem Schematic

In order to interface from the keypad to the MSP432, each of the 7 pins is connected to a general I/O pin on the MSP, which is displayed in Figure 6. As shown on Figure 7, each of the pins labeled P1-7 correspond to a row or column, with P2, P4, and P6 corresponding to columns and P1, P3, P5, and P7 corresponding to rows on the keypad. Each of the column connections contains a pull up resistor, which is used in the scanning process to determine which key was pressed.

Display

The game display and its relevant connections to the MSP432 can be seen below in Figure 8.

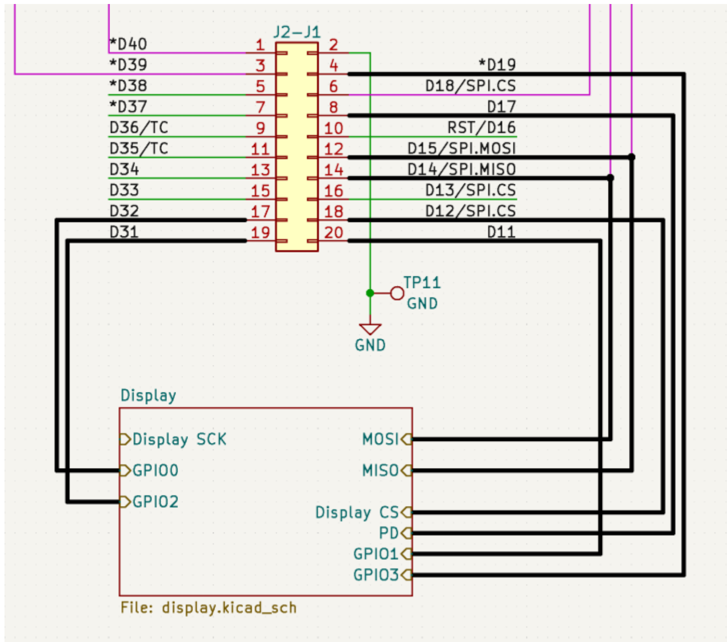

Figure 8. LED Display Hierarchical Schematic View

A subsystem view of the display can be seen below in Figure 9. The primary connections relevant to the electrical subsystem of the board are the 3.3V supply, which supplies VDD, the 5V supply that connects to VBL, and ground, which connects to VDD and VBL.

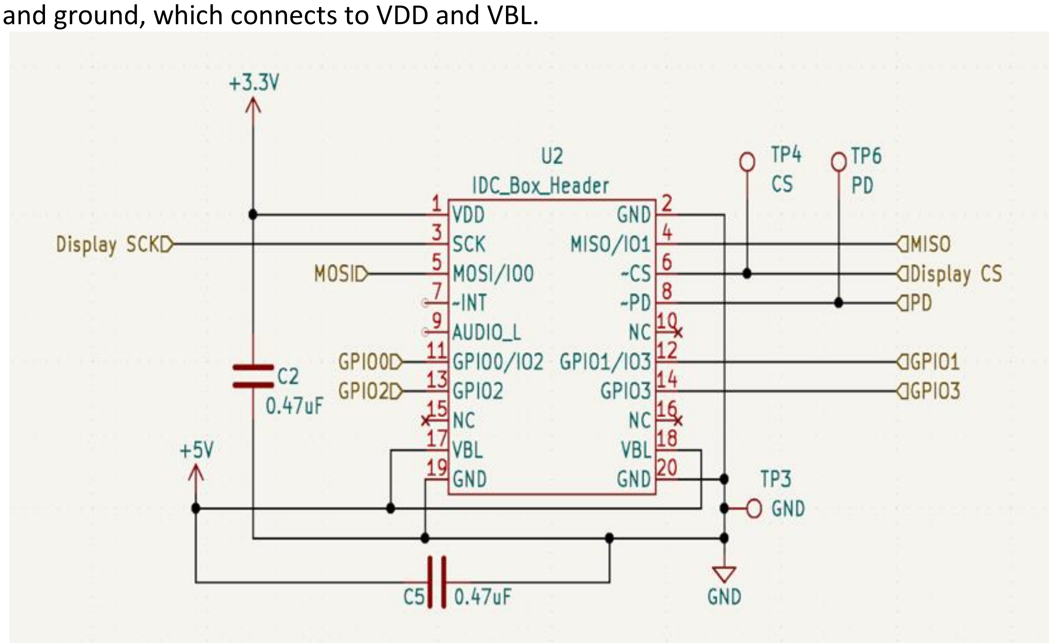

Figure 9. LED Display Subsystem Schematic

The necessary electrical test points were placed on the Display CS pin (TP4), the PD pin (TP6) and the ground supply (TP3). All the remaining connections not explicitly mentioned are relevant to the software aspect of the game console.

Table 3. Display Pin Connections

Connection	Function
VDD	Power Supply
Display SCK	SPI Clock (Input)
GPIO0	General Purpose IO0
GPIO2	General Purpose IO2
VBL	Input Voltage for LED Backlight Driver
GND	Ground
MISO	Main-In-Secondary-Out
Display CS	SPI Chip Select
PD	Power Down Control
GPIO1	General Purpose IO1
GPIO3	General Purpose IO3

MSP432 Microcontroller

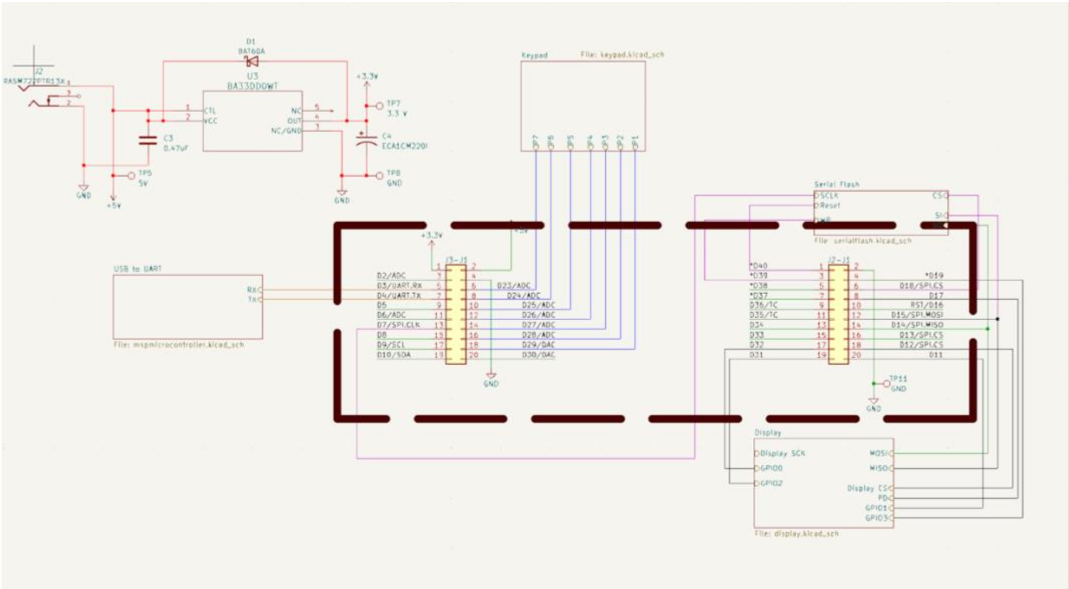

Figure 10. MSP432 Hierarchical Schematic View

As stated previously, the entire system functions through the MSP432, and can be seen in the dashed box in Figure 10. Each of the components relevant to the game function, such as the keypad, USB to UART COM port, serial flash, and display, all interface to the 20 I/O pin header board on the microcontroller.

PCB Layout

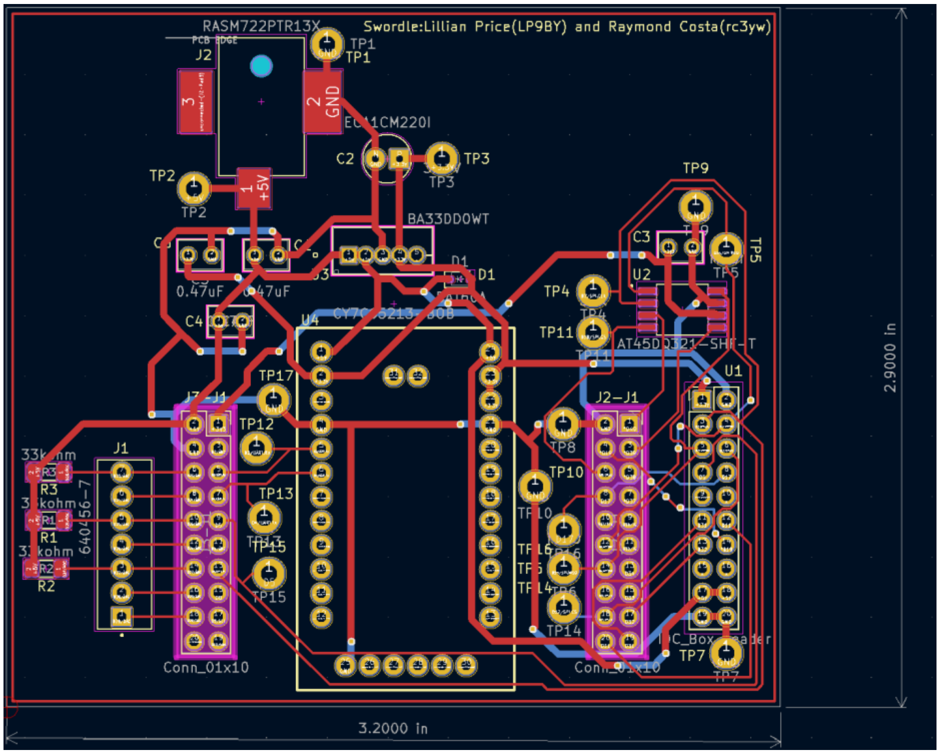

Figure 11. PCB Design Layout

The final PCB layout design can be seen above in Figure 11, which was built with the intention of mounting directly to the 20 header pins on the MSP432. The overall dimensions for the PCB were 3.2 in in width by 2.9 in in height, which was smaller than the display itself so it would easily fit within the game console enclosure. The first footprint placed in the layout process was the 20 header pins of the MSP423, which were placed at the exact same distance as on the microcontroller itself so that the MSP432 could be placed on top of the PCB. The remaining footprints were placed in order to be in close proximity to the other components they were electrically connected to, which would minimize having a complicated nest of wires. All the footprints were placed with the intention of being attached to the top of the PCB, excluding the voltage regulator which was the only component mounted to the bottom of the PCB. Finally, the footprint for the power jack was placed on the upper lefthand side of the board in order to easily feed the outlet plug through an opening in the enclosure to the power system.

Project Time Line

The initial proposed design schedule can be seen below in Figure 12 and the final design schedule appears in the following figure, Figure 13. In the design process, Lilian was the primary on all the hardware designs, which included schematics and PCB layout, and was secondary to the software development aspect of the project. Raymond served as the primary for the software development while serving as the secondary for any hardware development and testing.

One of the primary differences between the two proposed timelines came from the design and fabrication of the PCB. Initially, our group anticipated participating in the first and last session to order an initial and revised PCB, however due to initial delays with the start time of the project and some difficulties with the initial design, the two order dates were replaced by taking more time on the initial design and only shipping out one PCB board. The change to the PCB schedule can be seen in the final Gantt chart, and was responsible for the additional amendments in the schedule.

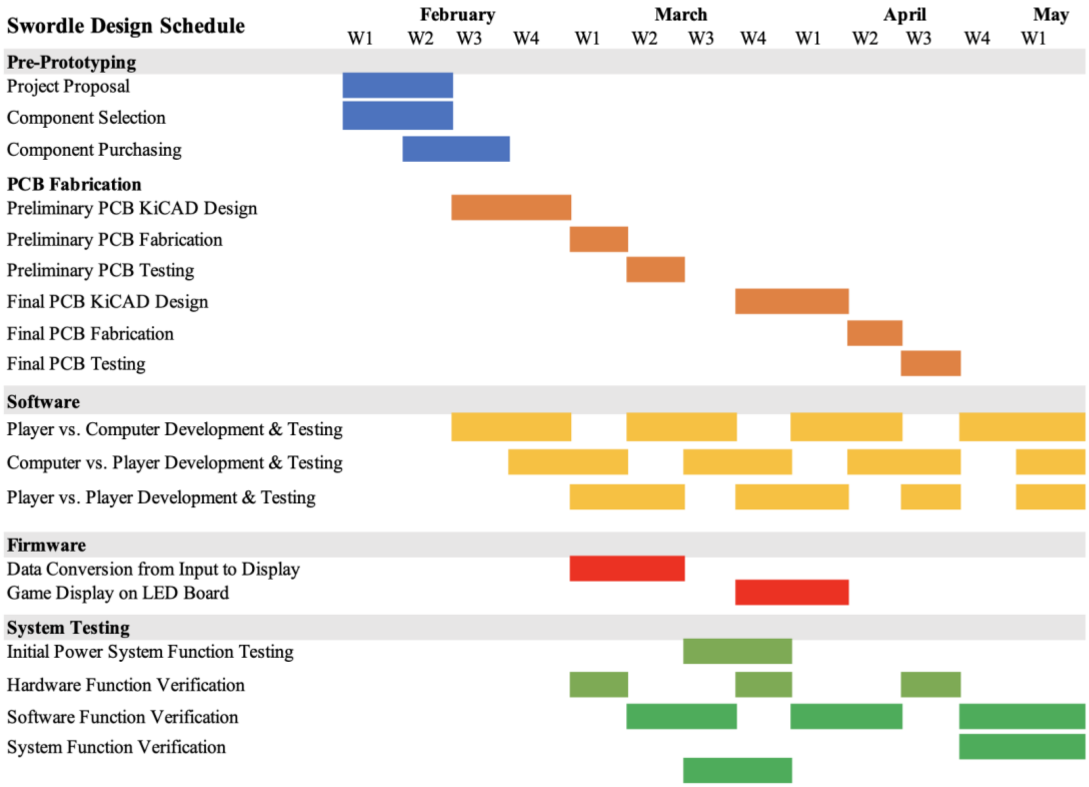

Figure 12. Initial Design Proposal Schedule

As shown below, making the necessary changes to the PCB design schedule pushed back our software design. After getting the PCB board itself, we experienced some delays in getting our board populated with our components, which made it difficult to begin testing the initial software functionality. Additionally, we ran into some electrical issues with our board so we experienced a delay in getting the board working since we had to make some changes ourselves. While experiencing the delays with our board we were able to get the basic functionality of the software set up to perform functions such as displaying the default screen on the display, or creating the scanning functions for the keypad. A majority of the software tasks were able to be performed in parallel with the PCB board construction, since we were able to verify certain functionalities within the terminal window of our code. Apart from what we could verify without a completed PCB board, we had to wait until the board was received and debugged in order to start testing the display, so that task was unable to be performed in parallel to the PCB board design.

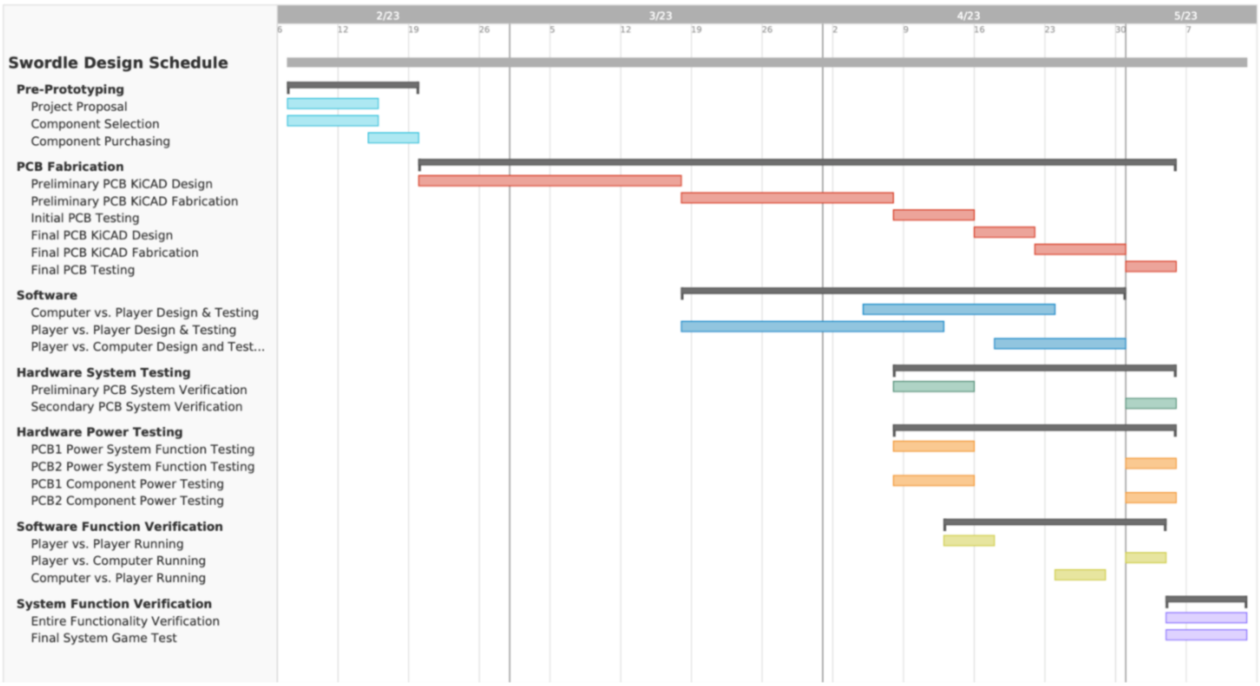

Figure 13. Finalized Design Schedule

Overall, the primary difference between the two versions of the design timeline is the PCB board design and fabrication process. Since we experienced quite a few setbacks in that process, our software development time was shortened significantly which made it difficult to achieve the full extent to what our hope was for our final deliverable on May 11th.

Software

In developing the software for our project first took a top down to designing our system. We first laid out the gameplay algorithm and everything it would entail order to identify the necessary subsystems that would need to be developed in order for the program at large to be successful. In doing so we identified three major subsystems that would need to have information sent to and from our MSP432: our keypad to supply user input, our serial flash to supply words and hold a dictionary of acceptable words, and our LED driver in order to properly communicate with our display, all of which are displayed below in Figure 14. In creating our final game mode player vs. player we used keypad inputs, in unison with feedback from our display based off of those inputs, in order for the game to be played.

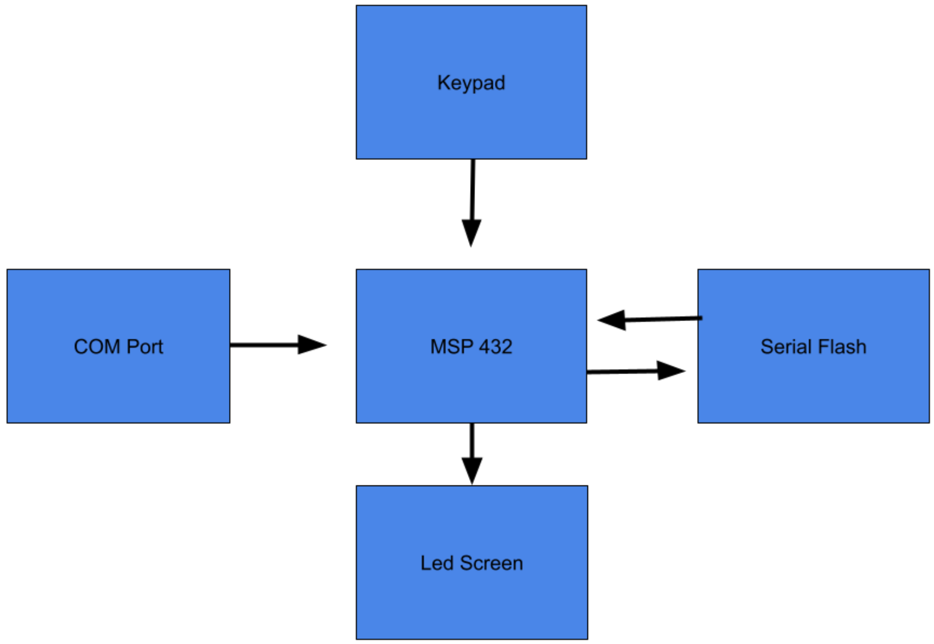

Figure 14. Software Subsystem Overview

Our keypad had 7 pins associated with each row and column of the pad. We connected each of these pins to general input output pins on our MSP432, which is shown in Figure 6. On our PCB board we placed pull up resistors of 33 kohms on each pin associated with a keypad column to our 5V source. Each pin on the MSP432 associated with a column was set as an input and pulled high externally. Each pin associated with a row initialized as an output and pulled high using internal pull up resistors within the MSP432. Once our pins were initialized, we used a function referred to as Scan() which output a char to determine if a key on the pad was pressed. The way this was accomplished was by setting a given singular row low and then checking if any of our columns, which were pulled high, were driven low. Each button on a given row if pushed would cause a switch to be activated and make a connection between the column and row pins for a given key. So, after a given row was set high, the function scan would check if the voltage on any of the three column pins was driven low. If any of them were, that would indicate that the key associated with that row and column had been pressed and a char value associated with that key would be returned. If no columns were driven low the row that was pulled would be internally pulled high and then the process would repeat with the next row being driven low. The logic behind the Scan() function is displayed below in Figure 15.

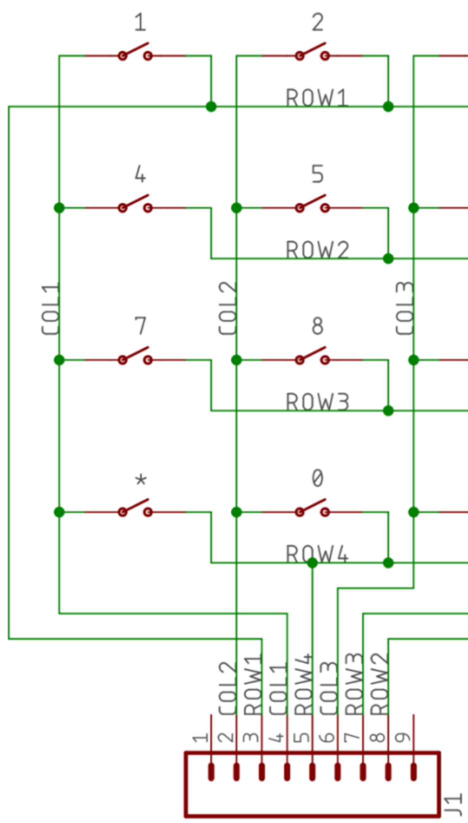

Figure 15. Keypad Logic

Our MSP432 used SPI in order to send and receive information to our serial flash memory. The MISO pin of our MSP432 was configured as an input pin, our MOSI pin was configured as an output pin, and our CLK pin was configured as an output pin. The chip select, reset, and write protect pins were configured as output pins and the interrupt pin was configured as an input pin. Each of these pins is associated with a component of the same name on our serial flash except for the MISO and MOSI which were connected to the opposite component on the serial flash.

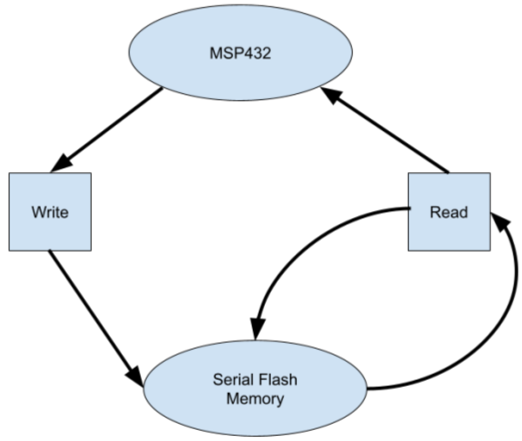

Figure 16. Serial Flash Logic

In order to communicate with the serial flash we configured our MSP432 to operate on a clock system with a rate of 12 MHZ SMCLK(sub main clock) and MCLK(master clock) to operate our SPI data flow. Our CS pin was driven low in order to initiliaze serial flash communications. We uploaded dictionaries to our serial flash using a function called dataflashMemoryProgramThruBuffer2WithErase() which would first send an 8 byte value to a given address in our serial flash by first sending the predefined opcode of 0x85 which signaled to the serial flash that data was being sent and that address was to be erased, followed by a 32 bit address, followed by the 8 bit data being stored in that address. This used a for loop in order to send the entire contents of a TX buffer which stored one byte units of data into the serial flash, which is demonstrated above in Figure 16. This would toggle the CS in order to send data on its rising edge, and each time a byte was sent the clock would be toggled as well. In order to read a full page worth of data from the MSP432 we used a function called dataflashMemoryPageRead() which would perform a similar function that of dataflashMemoryProgramThruBuffer2WithErase(), however a different opcode was provided to the serial flash and data was read on each toggle of CLK and CS on the MISO as opposed to sending data through the MOSI. We didn’t end up using our serial flash functions as we only implemented our player vs. player functionality of the game which takes a player inputted word and compares it to another player’s guess so we did not need to access our pre-defined dictionary.

For communication with our display, we used SPI as well, using the same MOSI, MISO, and CLK pins as our serial flash, but asserting a different chip select pin as low in order to send data to the display as opposed to the serial flash. This communication ran on the same internal clock of 12 MHZ provided by the MSP432, which was also used by the display’s internal graphics engine.

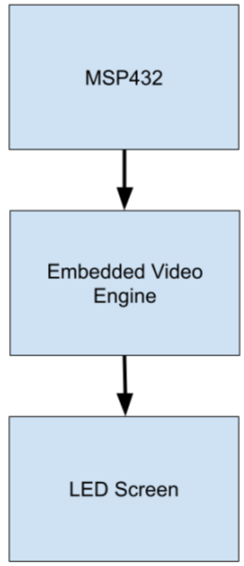

Figure 17. Display Logic

In order to initialize the display, we first had to toggle an pin that was resting high referred to as PD. This was configured as an output pin and controlled the on-off functionality of the display. When driven high the display was turned on. In order to communicate with the display, we used 3 different types of command functions. Two of the functions read and wrote to registers in the displays graphics engine by first sending the 32 address of the register preceded by a 2 bit command of 00 if it was read, 01 if it was to send a host command as opposed to communicate with a register, and 10 to send data to a register. To initialize the display, we first had to send two predefined command opcodes using the command function described previously to set the graphics engine to operate on the MSP432's clock system, and then send a dummy byte to activate. Then we primarily operated the display by updating a circular FIFO buffer stored in RAM_CMD. we used two commands, one which identified that a key was to be displayed, followed by the parameters of the key including width and a letter. The second command we used identified that there would be a color change which would send the color change opcode and then the hex value of that color. Additionally, we received data from the display on an interrupt pin which was defined as an input and driven low, if it were driven high a time delay was implemented until the display could receive more information.

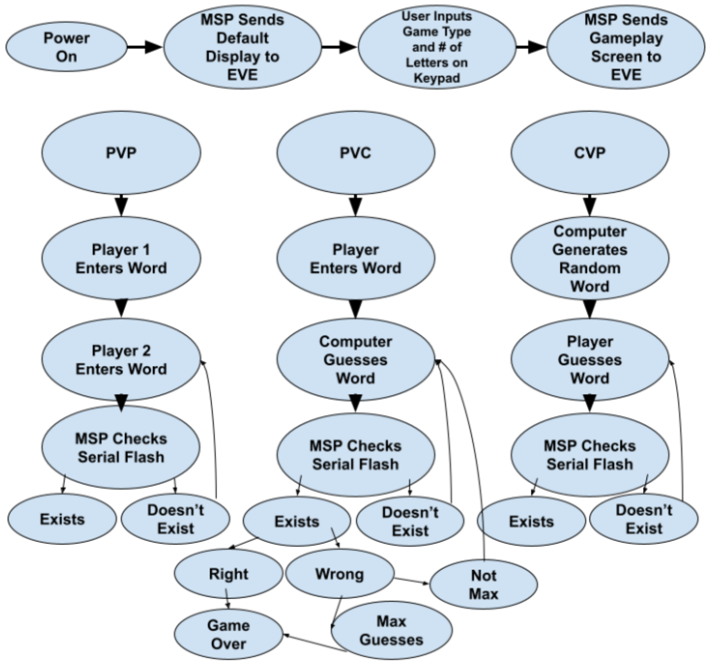

Figure 18. Game Logic

The logic of the game, which is visible above in Figure 18, was as such that if a given key was pressed the display would respond accordingly so that if you clicked up, down, left, or right it would move your location on the display which had been initialized using the keys and color function. Additionally, if you pressed enter it would store your letter in a word buffer and display your letter entry on the screen. This would repeat until you entered a full word which would then be stored as the word your partner would guess. Then the word would be cleared and the process would be repeated by scanning keys to determine if they were pressed and then relaying the appropriate response to the display. This would repeat until you either reached the allotted number of guesses or got the word correct in which the game would go back to the initial display screen and the process would start over.

Test Plan

In our original test plan design, we decided to break the system down between hardware and software, and then conduct identical tests in parallel to one another. Firstly, with the hardware system, the general idea behind the testing was to break it down into all the subsystems that were powered from the regulator. Initially the input voltage was tested using a DMM, which produced our expected 5V supply, and the output of the voltage regulator was tested and verified a 3.3V output supply. Before plugging any external components into the PCB, such as the display or keypad, which are high susceptible to damage if the voltage exceeds their maximum rating, we tested the voltage supply for each subsystem. Our testing consisted of using the test points that were added in the design phase to verify that each track was being supplied with the correct voltage.

After an initial verification, we plugged external components in one at a time and measured each node again to verify that our supplies. In doing so we encountered an error with one of the connections to the USB-UART COM Port. As seen in the image below, Figure 19, the 3.3V track that passes by pin 24 was turning into a 5V supply whenever the COM port was plugged in. Since in our design we placed the 3.3V wire too close to the 5V supply, we had to terminate that 3.3V connection in order to avoid damaging the COM port while still being able to use it. The process of placing slowly populating the board with external components while verifying the supplies ultimately lead us to a fully functioning power system.

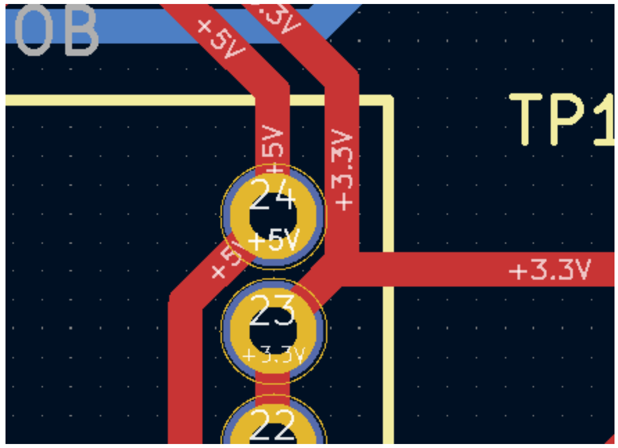

Figure 19. Wiring Error on 3.3V Supply to COM Port

Once the power system was verified, the same approach was used for testing the software. As demonstrated in the figure below, the software testing consisted of using an iterative process for each subsystem to verify functionality. Each subcase to test was marked with a physical test point on the PCB board.

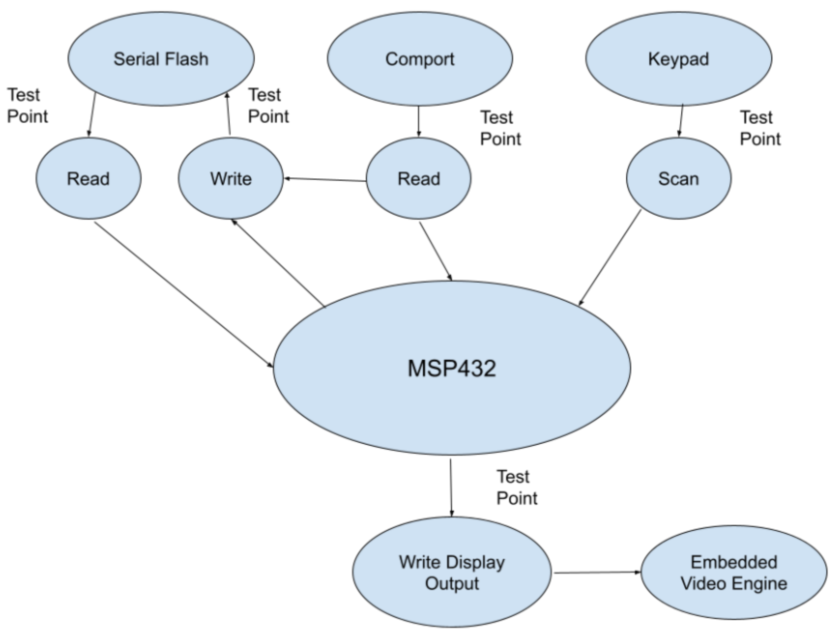

Figure 20. Software Testing Hierarchy

The first subsystem consisted of the serial flash, which could be tested in verified in the absence of the PCB itself. This testing process consisted of reading and writing test data values to and from the serial flash via the SPI. In a similar fashion to the serial flash, the keypad was also tested in the absence of the PCB board by pressing a key and having it display within the terminal window. Finally, the last component to test was the communication from the COM port to the serial flash, via the MSP432. This aspect of the project experienced the most change during our test planning. After taking into account the electrical connectivity problems and evaluating what we believed we could accomplish with the software, our group decided to exclude the COM port from our design. Although we had some basic capabilities working for the COM port, that branch of testing was removed from our original design plan. After verifying both the serial flash and keypad, the entire system was tested using the LED display.

Ultimately, the final test was conducted by playing a run through of the player vs. player mode of the game. By integrating all the subsystems together, that would verify if our previous testing was properly conducted. During our final test we were able to successfully play multiple rounds of player vs. player.

Final Results

Overall, our Swordle game console was able to successfully play successive rounds of player vs. player for each letter option, ranging from 2-5 letters. The first player was able to not only select the game mode and number of letters, but they were also able to input a word. The opponent was able to successfully choose words, and was provided with correct feedback from the screen for different guesses. When the player would correctly guess the chosen word, or run out of guesses, the display would return to the home screen and another round could begin, so the system was compatible with playing multiple rounds.

In terms of design criteria, our overall grading system was based on the various player modes, with all three functioning corresponding to the highest possible grade, two functioning corresponding to a median grade, and one functioning corresponding to the lowest grade of the three. One of the consistent criteria in each grade range was functionality in terms of the electrical system and user interface. We were able to successfully get our screen to display and respond to prompts in addition to getting a fully functioning keypad that could communicate to the display.

When considering the overall proposed design of three different game modes, our group was unable to successfully implement all three versions of the game. Although we were able to complete the base functions needed for the two remaining game modes that used a computer opponent, we were unable to finish the game rules in order to successfully demonstrate computer vs. player and player vs. computer mode. Therefore, as outlined by our grading rubric, our project would fall within the C range as outlined below.

Letter Grade:	**Design Criteria:**
A	● Design correctly functions for multiple rounds of **all versions** of the game: ○ Player vs. Computer ○ Computer vs. Player ○ Player vs. Player ● LED display is fully lit and readable for entirety of game play ● LED display responds to user input ● LED displays prompts and user input

<table>
<tr><td></td><td>Game Instructions are easily understood and device is easy to use
Device is professional in appearance</td></tr>
<tr><td>B</td><td>Design correctly functions for multiple rounds of at least 2 versions of the game:
Player vs. Computer
Computer vs. Player
Player vs. Player
LED display is fully lit and readable for entirety of game play
LED display responds to user input
LED displays prompts and user input
Game Instructions are easily understood and device is easy to use
Device is professional in appearance</td></tr>
<tr><td>C</td><td>Design correctly functions for multiple rounds of at least 1 version of the game:
Player vs. Computer
Computer vs. Player
Player vs. Player
LED display is fully lit and readable for entirety of game play
LED display responds to user input
LED displays prompts and user input
Game Instructions are easily understood and device is easy to use
Device is professional in appearance</td></tr>
<tr><td>D</td><td>Design correctly displays input characters on the screen
LED display is fully lit and readable
LED displays prompts
Device is professional in appearance</td></tr>
<tr><td>F</td><td>Device turns on and lights up the display
Game Instructions are easily understood
Prototype is present and professional in appearance</td></tr>
</table>

Costs

Overall, our team stayed well within our allotted budget of $500 for the entire project. A detailed list of the cost breakdown can be found below in Table 4, and a more detailed itemized list can be found in Appendix A.

Table 4. Bill of Materials

Item:	**Quantity:**	**Cost/Item ($):**	**Total ($):**
Display Screen	1	143.32	143.32
MSP432	1	0.00	0.00
PCB Board Assembly	1	33.00	33.00
USB to UART (COM Port) and Breakout Board	1	20.69	20.69
Serial Flash	1	4.3	4.3
Keypad	1	4.95	4.95
Voltage Regulator	1	3.49	3.49
Electrolytic Capacitor	1	0.28	0.28
Passive Capacitors	4	0.35	1.4
Resistors	3	0.1	0.3
Through hole Connectors	2	0.39	0.78
Power Jack	1	2.48	2.48
Test Points	16	0.42	6.72
External Wall Mount	1	9.39	9.39
Budget:	----------	---------------	**500**
Remaining Fund:	---------	---------------	**268.9**
Total:	---------	---------------	**231.1**

As shown in the cost breakdown above, the most expensive component in this design is the display screen. In terms of breaking down cost for the overall system, it would be quite difficult given that typically there are no more than 200 screens in stock at a time, and the quantity of the order does not decrease the unit price of $143.32. In terms of the remaining components, their cost is already quite low, but ordering in units of 1,000 would make some slights reductions in the unit price for some of the more expensive components such as the serial flash which going from an individual purchase to 1,000 units decreases the unit price from $4.18 to $3.02. Additionally, the other most expensive component, which is the external wall mount, would have a significant decrease in unit price if it were ordered in 1,000 units as opposed to a smaller quantity. Ordering 1,000 units of the external wall mount would decrease the unit price from $9.39 to $6.57, which is quite a significant jump. Overall, small decreases in the overall cost are possible for the smaller components, but the largest cost factor, the display screen, would not change even if the quantity of the order were significantly higher.

Future Work

Throughout the design process, our group encountered numerous difficulties in relation to both the hardware and software aspects of the system. Although the hardware system is fully functioning correctly, correcting the errors with the design and printing a new board would be a strong first step in improving the project. Since we had our mount to the MSP432 header board on backwards, our group had to create individually wired connections between the MSP432 and the display. Additionally, we had an electrical error from a wire that was placed too close to a component on the COM port, which cause a 3.3V supply to jump to 5V when the COM port was plugged in, so adding spacing would correct that issue as well. In order to save time later when focusing on software and due to the few number of changes needed to the PCB board, it would be beneficial to fabricate a final PCB board for the design.

Overall, the primary focal point moving forward in the software would be to implement the game logic for the player vs. computer and computer vs. player game modes. Our group was able to complete the base functionality that was necessary to implement those game versions, such as communicating with the flash, but we were unsuccessful in finishing the game design in time for the final deliverable. Given that the storage on the serial flash is quite large at 32 Mbits, it is possible to expand the game to include more libraries that could potentially span multiple languages. Additionally, small changes could be made in regards to the professionality of the display screen, adding additional game features, or expanding the number of available letters. The expected errors in software in this case would relate primarily to minor bugs in designing the separate game rule files, which should not pose too daunting of a challenge given that the display screen is also extremely useful for debugging.

References

[1] “Wordle - A daily word game.” https://www.nytimes.com/games/wordle

[2] United States Congress Senate Special Committee on Aging and United States Congress Senate, “Combating Social Isolation and Loneliness During the COVID-19 Pandemic: Hearing Before the Special Committee on Aging, United States Senate, One Hundred Sixteenth Congress, Second Session, Washington, DC, June 11, 2020,” U.S. Government Publishing Office, Washington, 2022. [Online]. Available: https://purl.fdlp.gov/GPO/gpo176494

[3] “KiCad | 6.0 | English | Documentation | KiCad.” https://docs.kicad.org/6.0/en/kicad/kicad.html

[4] "FreeDFM," Advanced Circuits, [Online]. Available: https://www.my4pcb.com/net35/FreeDFMNet/FreeDFMHome.aspx. [Accessed 11 May 2023].

[5] WWW Electronics Incorporated, Charlottesville, VA, 2023.

[6] “CCSTUDIO IDE, configuration, compiler or debugger | TI.com.” https://www.ti.com/tool/CCSTUDIO

[7] MSP430 microcontrollers | TI.com.” https://www.ti.com/microcontrollers-mcus-processors/msp430-microcontrollers/overview.html

[8] I. Wojnowska-Baryła, K. Bernat, and M. Zaborowska, “Plastic Waste Degradation in Landfill Conditions: The Problem with Microplastics, and Their Direct and Indirect Environmental Effects,” Int. J. Environ. Res. Public. Health, vol. 19, no. 20, p. 13223, Oct. 2022, doi: 10.3390/ijerph192013223.

[9] J. Jackson, “No e-wasteland for electronic waste disposal: effective legislation to protect communities surrounding landfills,” Journal of Gender, Race and Justice, vol. 18, no. 2, p. 499+, 2015.

[10] “16 CFR Part 1505 -- Requirements for Electrically Operated Toys or Other Electrically Operated Articles Intended for Use by Children.” https://www.ecfr.gov/current/title-16/chapter-II/subchapter-C/part-1505

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

[12] “Specification for base materials for rigid and multilayer printed circuit boards ” IPC [Online]. Available: https://www.ipc.org/TOC/IPC-4101C.pdf. [Accessed: 11 May 2023].

[13] JEDEC, SMT / SMD Components & packages, sizes, dimensions, details.

[14] D. M. Threlkel, “NEMA Enclosure Types,” no. 17.

[15] “1910.304 - Wiring design and protection. | Occupational Safety and Health Administration.” https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.304

[16] “16 CFR Part 1505 -- Requirements for Electrically Operated Toys or Other Electrically Operated Articles Intended for Use by Children.” https://www.ecfr.gov/current/title-16/chapter-II/subchapter-C/part-1505

[17] L. M. Benson and K. K. Reczek, “A Guide to United States Electrical and Electronic Equipment Compliance Requirements,” National Institute of Standards and Technology, Gaithersburg, MD, NISTIR 8118r2, Jun. 2021. doi: 10.6028/NIST.IR.8118r2.

[18] Adrian E. Aukens. "Word Game". United States of America Patent US3355172A, 1965.

[19] Sidney Jay Hammer. "Apparatus for Word Guessing Baord Game". United States of America Patent US0261554A1, 2006.

[20] Melford D Clark. "Word Game Device". United States of America Patent US3411221A, 1965.

Appendix

Index	Manufacturer Part #	Digikey Part #	Qty Req'd	Per Unit Price	Cost
1	FG14X7R1E474KNT06	445-180818-1-ND	4	0.35	1.4
2	ECA-1CM220I	10-ECA-1CM220ICT-ND	1	0.28	0.28
3	BAT60AE6327HTSA1	BAT60AE6327HTSA1CT-ND	1	0.5	0.5
4	640456-7	A19472-ND	2	0.39	0.78
5	RC1206FR-0733KL	311-33.0KFRCT-ND	3	0.1	0.3
6	5011	36-5011-ND	11	0.42	4.62
7	5012	36-5012-ND	1	0.42	0.42
8	5010	36-5010-ND	1	0.42	0.42
9	5014	36-5014-ND	1	0.42	0.42
10	5013	36-5013-ND	1	0.42	0.42
11	5126	36-5126-ND	1	0.42	0.42
12	NHD-7.0-800480FT-CSXV-CTP	NHD-7.0-800480FT-CSXV-CTP-ND	1	143.32	143.32
13	AT45DQ321-SHF-T	1265-1293-1-ND	1	4.3	4.3
14	BA33DD0WT	846-BA33DD0WT-ND	1	3.49	3.49
15	CY7C65213-BOB	448-CY7C65213-28PVXITCT-ND	1	4.18	4.18
16	RASM722PTR13X	SC2028-1-ND	1	2.48	2.48
17	COM-14662	1568-1856-ND	1	4.95	4.95
18	BOB-13830	1568-1504-ND	1	16.5	16.5
19	16-00066	839-1669-ND	1	9.39	9.39

Figure 21. Full Bill of Materials