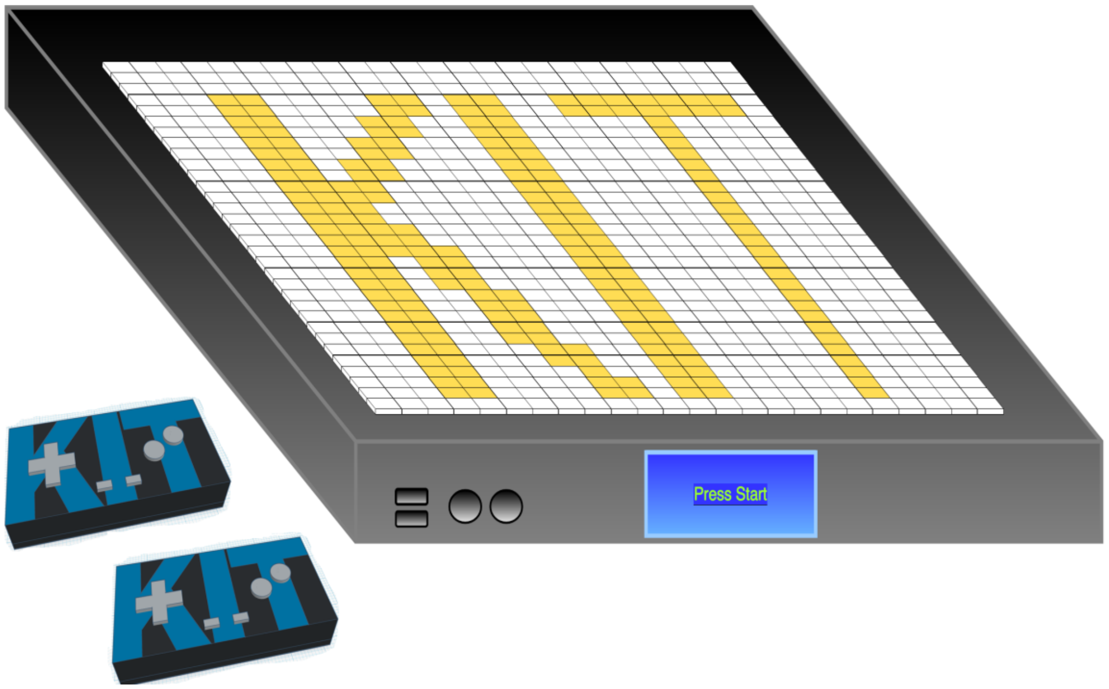
SENIOR DESIGN 1
K.I.T.
Knight Interactive Timebox



University of Central Florida
Department of Electrical Engineering and Computer Science
Dr. Lei Wei

Group 17

Jeremy Hughes	Electrical Engineer
Ronald Lugo	Electrical Engineer
Michael Ruckstuhl	Computer Engineer
Oraya Vachirachaiyakarn	Electrical Engineer

Table of Contents

Table Lists

Table of Figures

1. Executive Summary

Interactive entertainment is a major part of today's entertainment industry. The developers of video games are trying to stay ahead of the sales curve by adding new control and display capabilities while rehashing the same basic stories, strategies, and gameplay types seen time and time again. It is becoming more apparent that complicating game mechanics does not lead to success. As games become more complex old school gamers are losing interest and looking back at their retro gaming consoles. The Knight Interactive Timebox (K.I.T.) simplifies display and control capabilities while taking advantage of the retro gaming market. Essentially the K.I.T. is comprised of an LED display, controllers for user inputs, and a microcontroller to operate display and game software.

The K.I.T. simplifies the game display and development by utilizing the following components. A raspberry pi 2 is used to handle game logic. The raspberry pi computation system series have long been used to emulate classic gaming platforms such as the Nintendo Entertainment System, Sega, PlayStation and many more. The Raspberry Pi system has more than proven its versatility in classic gaming capabilities. The beauty of using this system is the many different coding languages that are compatible with the Raspberry Pi. Developers are given freedom in coding languages such as Python, C++, and Java among others. This allows developers to create games without limitations.

To ensure that the user receives a simple game the display resolution is restricted to a 24 x 32 LED matrix. This matrix allows easy coding as the resolution is a multiple of 8. This restricts complexity of game types by reducing the visual cues capable for each game type. Finally, K.I.T. simplifies game capabilities by using a classic input controller architecture. With game inputs constrained to four directional inputs and two selective inputs, the gamer is not overwhelmed by cumbersome input gameplay.

1.1 Marketing Ideology

The technological advancement in computation capabilities during the late 20th century brought dramatic new experiences for society. Systems with sophisticated computation capabilities were becoming available to the general public. With personal computer capabilities rapidly advancing along with the development personal gaming consoles such as the Nintendo Entertainment System or the ATARI, a new era of family interaction. Along with the advancements in gaming came the growth of new social circles that shared their passion. Whether it was finally defeating that first mushroom in Nintendo's Super Mario Bros. or witnessing your entire party die of dysentery in feeble attempts at MECC's The Oregon Trail, many young adults share fond memories of their first gaming experiences. However, it seems as gaming becomes more complex there is a barrier dividing the new generation of gamers from the older generations.

More often it seems that parents are becoming disappointed in their children's gaming habits while previous generations shared the controller with their father or asked their mother to help them beat that particularly challenging level. One of the factors driving this change is the complexity of today's gaming platforms. Early video game designs received few inputs from the player, relied on low resolution displays, and implemented simple strategies for success. If only there was some device that brought back the simplicity of these classics allowing gamers of all generations to connect through video game entertainment. These ideologies are the motivating factors for the design of the K.I.T..

The gaming market may have ebbs and flows resulting from saturated markets and new technological advancements but over the last thirty years it has grown considerably. For the fiscal year of 2017, Digital Trends stated "Total spending for interactive entertainment across all devices and platforms was $91 billion" [78]. Retro gaming still has a major piece of that market which was made clear by the gaming giant Nintendo.

In June of 2018 the NES Classic Edition "outsold heavyweights like the PlayStation 4, Xbox One, and even Nintendo's own Switch" [79]. With unit sales topping over three million in the span of three years Nintendo proved there is money to be made in retro gaming platforms. Retro game development is not fully understood without a discussion about one of the most well-known classics, Tetris. Tetris is a simple puzzle game designed for computers far less capable than what is currently available, and yet seems to be timeless. The player must manipulate puzzle pieces with only four main inputs. Both the display resolution and processing power required to play are extremely low compared to today's standards. Alexey Pajitnov created Tetris while working for the Dorodnitsyn Computing Centre of the Soviet Academy of Sciences. Box Brown depicted the development of Tetris in his graphic novel titled "Tetris: The Games People Play".

In reference to the game's creator Brown stated "He was doing this just to see if he could do it" [80]. Knowing the game's origin, it may seem surprising to learn that Tetris holds the record for highest unit sales of any video game. Topping the charts with over 170 million copies sold, Tetris far surpasses the most popular video games coming from more recent developers [81]. Tetris is a prime example that the complexity of a game does not define its success. The display and control capabilities do not have to be complex for a successful gaming platform.

1.2 Marketing Goal

The goal of the K.I.T. is to limit the resolution of the game display, the complexity of the controller, and the computational power of the CPU to guide game developers to rely on simple mechanics for entertainment success. Simplifying game designs open the consumer market to more than just the dedicated gamers. Classic gamers enjoy the nostalgic feel of the video games while new generations look forward to showcasing their gaming supremacy in a level playing field. Simplifying the design drastically reduces cost of the development process which in turn reduces the cost to the consumer.

1.3 Marketing Summary

The K.I.T. relies on a few simple functions. Much of the marketing requirements are outlined in section 2. Tables 1, 2, and 3 break down the different requirements for the design. The house of quality displayed in Figure 1 details how marketing requirements match up to engineering requirements. The most basic game board version will have a color display made up of 768 LEDS, as resolution of 24 X 32. The main user inputs will come from handheld controllers with four directional buttons and two selection buttons.

The game board will be capable of receiving inputs from four different controllers. The basic game board version will have memory for storage of 3 different game types. The game board will have memory read and write capabilities utilizing a USB interface. The game board will be capable of DC battery power operation as well as hardwired 110/220 volts at 60 HZ power operation with an integrated AC to DC converter (normal household capabilities). Finally, the game board will have a built-in speaker for audio capability.

2. Project Description

The overall description of K.I.T. such as project requirements, specifications, and constraints can be found under this section. The house of quality diagram is shown under section 2.3 where the marketing and engineering requirements are identified. The hardware and software diagrams are also included along with the initial budgeting table.

2.1 Project Requirements and Specifications

Listed below are the requirements and specifications for K.I.T. including the requirements for the LED board itself, the handheld controllers, and the software. Table 1 contains the lists of requirements for the LED board.

Table 1 LED Board Requirements

1.1	The device shall have the ability to be interactive with users
1.2	The LED board shall be less than 80 cm by 60 cm in size
1.3	The LED board shall be pixel-like displayed
1.4	The LED board shall be at least 16 x 32 LEDs
1.5	The LEDs shall be RGB multicolor
1.6	The LED board shall be hardwired-powered
1.7	The LED board shall be able to receive inputs from controllers
1.8	The LED board shall be able to receive and implement software from an external source
1.9	The LEDs shall have the ability to adjust brightness
1.10	The LED board shall be lightweight
1.11	The device shall have a small LCD screen component to display information to the user and developer
1.12	The LED board shall have a blue, black, and red color scheme

Table 2 shows the requirements for the handheld controller which dictates how the user interacts with the K.I.T.

Table 2 Controllers Requirements

2.1	The controllers shall have the capability to communicate to the board
2.2	The controllers shall have 2 buttons for selection purposes and 4 buttons to control directions (up, down, left, and right)
2.3	The controllers shall be battery-powered
2.4	The controllers shall be large enough to be held by two hands and small enough to not be bulky
2.5	The controllers shall be able to maintain charge for an acceptable period of time
2.6	The controllers shall be able to be easily packaged with the system
2.7	The controllers shall not be software specific and be able to be used with whatever software is running on the system

Table 3 include the software and gaming requirement for K.I.T. dictating what the software and hardware must be capable of.

Table 3 Software Requirements

3.1	The software shall have the ability to communicate to the LED board
3.2	The software shall consist of up to 3 games, which can be played up to 2 players at a time
3.3	All software shall be maintained in a group repository in order to ensure that there are backup copies
3.4	One game software produced shall be a matching game
3.5	One game software produced shall be a pong game with two player functionalities
3.6	One game software produced shall be a dungeon game
3.7	All software shall be tested to ensure functionality
3.8	Software shall be written in Python, C, or Assembly
3.9	All software shall maintain a commented header section detailing who wrote the software, what the software is for, and a simple changelog
3.10	All software variables shall follow CamelCase naming convention for all variables
3.11	All software variable names should be descriptive to their function

Table 4 details the testing requirements. It is important that the K.I.T. meets all testing requirements as they are directly tied to a successful implementation of the design. This requirement table is an overview of how the testing must be approached. Component testing is detailed further in its respective section.

Table 4 Testing Requirements

4.1	All tests should be verifiable with something to prove that they have been performed i.e. a picture or by marking the completion of the test on a test form
4.2	Each testable component shall have a maintained list of all tests that have or will be performed on the component to aid in replication
4.3	Tests shall be repeatable and verifiable for each component
4.4	Any and all testing shall be repeated on any replacement component for the project
4.5	All tests must be passed before the project is completed
4.6	All tests shall be clear about their purpose and expected outcomes
4.7	All components must be tested for functionality and integration
4.8	The final phase of testing shall be the comprehensive unit testing

2.2 Possible Project Constraints

Since this project is not being funded through any grants or corporations, money is a major constraint on the project. The project's current budget is only $800. Each member is expected to be able to fund the project with $200 of their own money. If project costs overshoot the projections, the financial burden on the group might be too great and lead to a reduction of scope of the project or a selection of different materials. During the project design, cost of components was an important consideration that influenced the choice of materials. As the project continues and more components are being bought and tested, budgetary constraints will become more and more prominent.

Another important constraint to consider is time. Since there are only four team members, there are only so many man-hours of work that can reasonably be expected from each teammate. The time must be considered for how much work each component of the project introduces. Construction time is also a time constraint to consider. The PCBs take time to order, make, and be received. This limits the possible number of revisions of the board that can be made. Consideration for how much time it will take to properly test and integrate all the components is another concern. Each of these issues show how much of a constraint time is on the project. This makes each week of senior design two critical to ensure success. Maintaining a schedule to ensure everything is completed on time will be important. Otherwise, the project could get rushed and/or functions could get cut from the project.

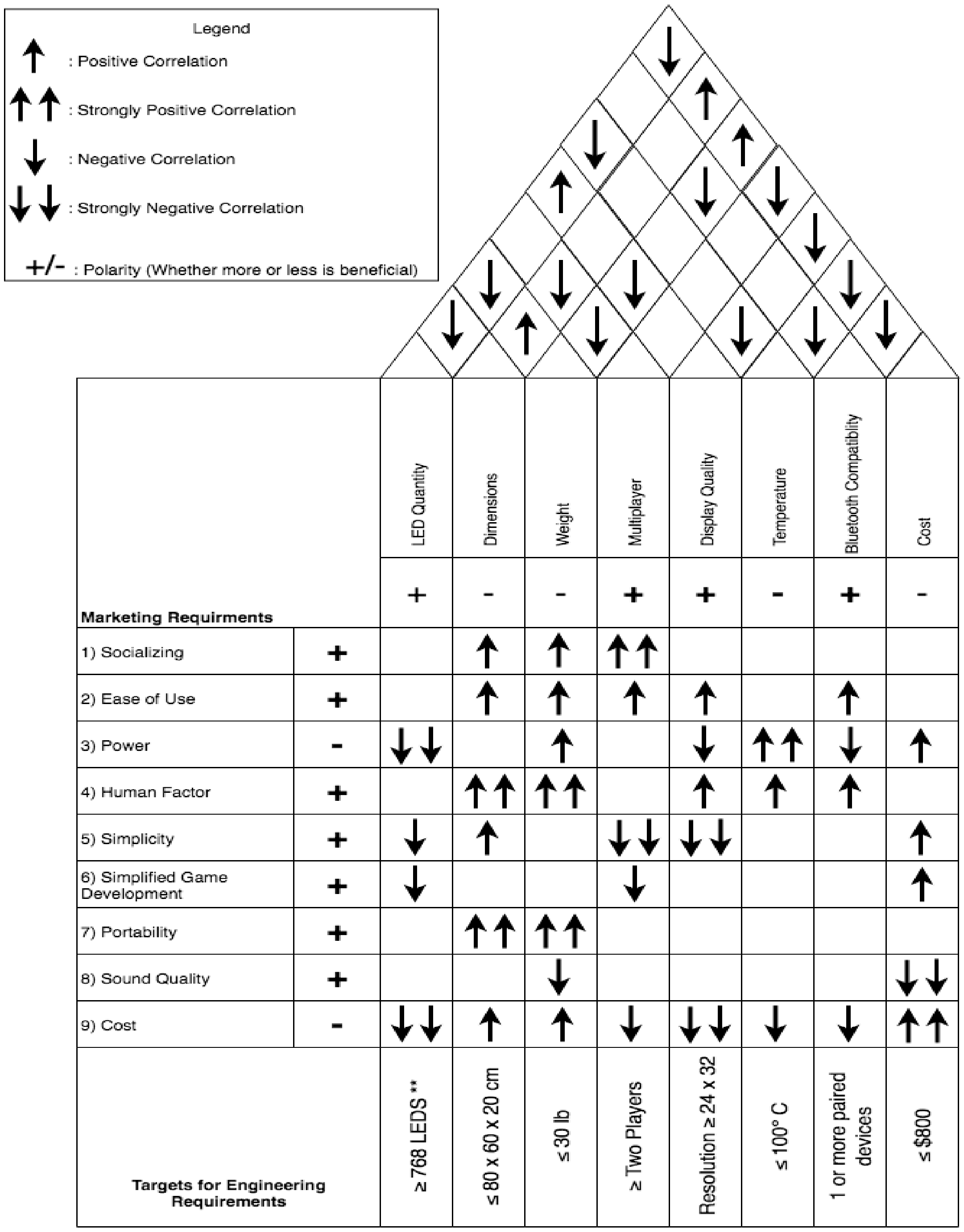
Another constraint will be the components of the project. Components must be researched, bought, ordered, and available. There have been several components that would have been useful to the project that were not selected due to them being unavailable. The components can act as a constraint to the project if they aren’t available and when they are chosen. The specifications of each component dictate what we can do with them. As the project continues and more components are bought and tested, we can no longer simply switch out components. If we need to change or add functionality to the project later on down the line, we will be limited by the components we have already selected. This type of consideration was important when selecting which components to choose. This led us to choose components that had more capabilities than we needed to ensure any changes further on in development could be accommodated.

The design is another constraint to the project. Since the design phase of the project is mostly done, we must follow the requirements and project outline we have already submitted. This limits us to how much we can change any features further on in development. The design might change slightly as we continue to develop the project and test components. The design requirements we have already submitted act as a constraint onto how much we can remove or add functionality at a later date.

Software can be another constraint to the project. The software must follow the outlined requirements such as multiplayer functionality. As the hardware is finalized, the software for each component must be ready as well. Learning the software specifications for each hardware component can take time. Similarly, ensuring each component’s software has the ability to perform properly is another consideration and will take both time and effort. Possibly learning new computer languages to write for the software can affect the time needed to develop the software. The software is an important factor in the project and must be focused on as a constraint.

2.3 House of Quality Diagram

Figure 1 below displays the House of Quality diagram. In the House of Quality, the features of the K.I.T. are compared with the market requirements shown on the left and the engineering requirements on the bottom. Finally, the "roof" of the diagram shows the correlation between each characteristic. The house of quality is used to better understand how changing in components effect meeting the desired requirements. For the K.I.T. the main requirements focused on are the display resolution, number of players, simplicity of game play, power consumption, and cost. As shown, there are more concerns than those four stated. One example of how changes in one component can affect a marketing requirement. Increasing the display size will allow the user to interpret more visual cues in the game. However, this can allow for more complicated game designs which is detrimental to simplicity. Increasing the display size will increase the number of LEDs used which is beneficial to meeting the engineering requirements, however, this also increases power consumption which is detrimental to the marketing requirements. Adding Bluetooth connectivity can make the system easier for social environments to meet the marketing requirement of sociability. However, adding Bluetooth capabilities will increase cost, power consumption, and weight. The K.I.T. is designed to be portable within reason. Easy to take from location to location. The video display frees the user from needing a video monitor to use. However, the larger the display is the more LEDs it requires. The more LEDs required the greater the power consumption and thermal output is. All of these factors also influence the cost of the device.


Legend
↑ : Positive Correlation
↑↑ : Strongly Positive Correlation
↓ : Negative Correlation
↓↓ : Strongly Negative Correlation
+/- : Polarity (Whether more or less is beneficial)

Marketing Requirments		LED Quantity	Dimensions	Weight	Multiplayer	Display Quality	Temperature	Bluetooth Compatibility	Cost
		+	-	-	+	+	-	+	-
1) Socializing	+		↑	↑	↑↑				
2) Ease of Use	+		↑	↑	↑	↑		↑	
3) Power	-	↓↓		↑		↓	↑↑	↓	↑
4) Human Factor	+		↑↑	↑↑		↑	↑	↑	
5) Simplicity	+	↓	↑		↓↓	↓↓			↑
6) Simplified Game Development	+	↓			↓				↑
7) Portability	+		↑↑	↑↑					
8) Sound Quality	+			↓					↓↓
9) Cost	-	↓↓	↑	↑	↓	↓↓	↓	↓	↑↑
Targets for Engineering Requirements		≥ 768 LEDS **	≤ 80 x 60 x 20 cm	≤ 30 lb	≥ Two Players	Resolution ≥ 24 x 32	≤ 100° C	1 or more paired devices	≤ $800

Figure 1 House of Quality

2.4 Hardware Block Diagram

Figure 2 below displays the Hardware Block Diagram of the K.I.T. design. This diagram is a brief summary of the different hardware components and their interconnections.

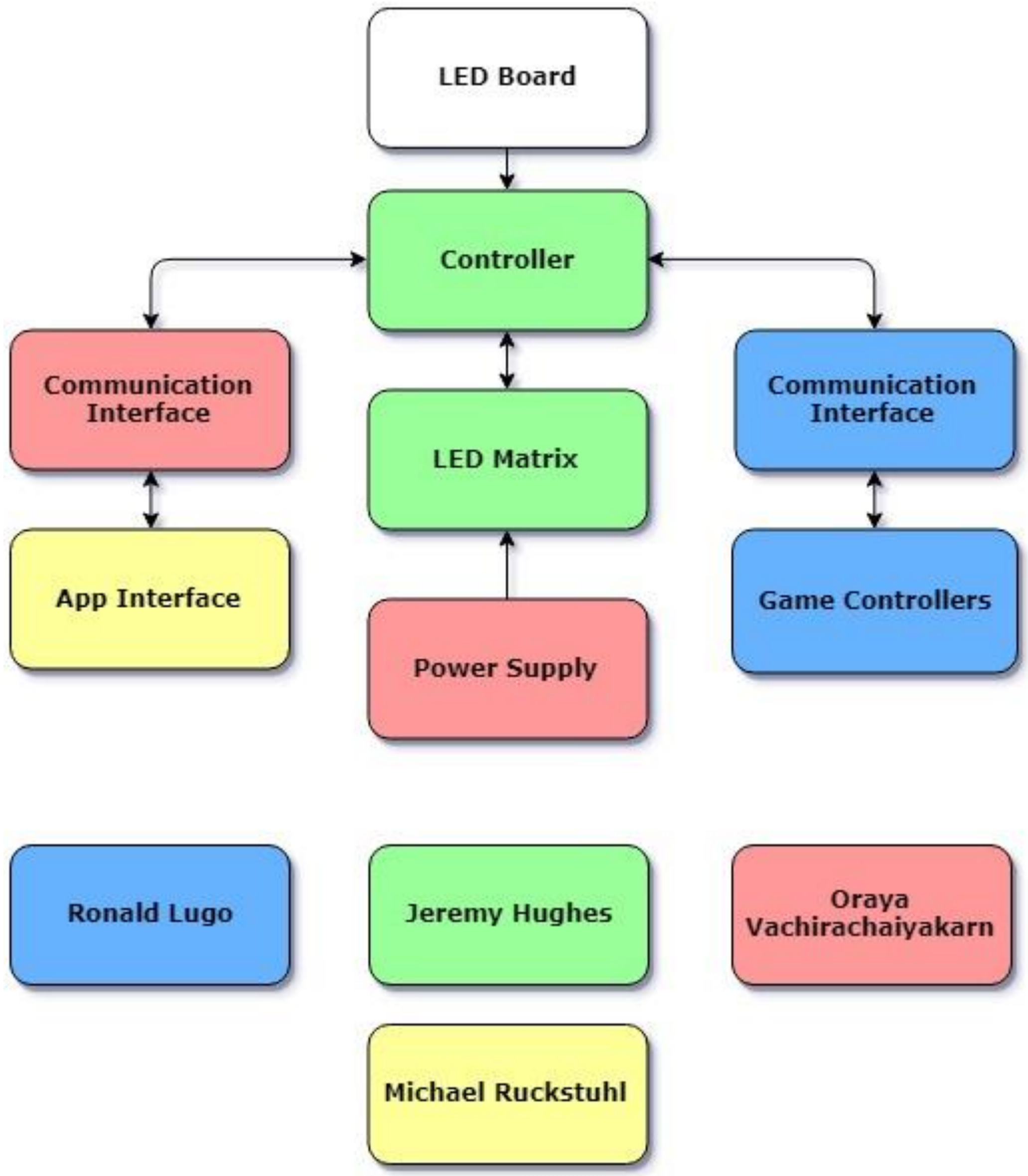


Figure 2 Hardware Diagram

2.5 Software Block Diagram

The general software diagram as shown in Figure 3 shows how each component will work together. The components will each have their own software associated with them that will talk to the other components of the K.I.T. project as shown by the diagram. The diagram is colored to show which team member will focus on which software component.

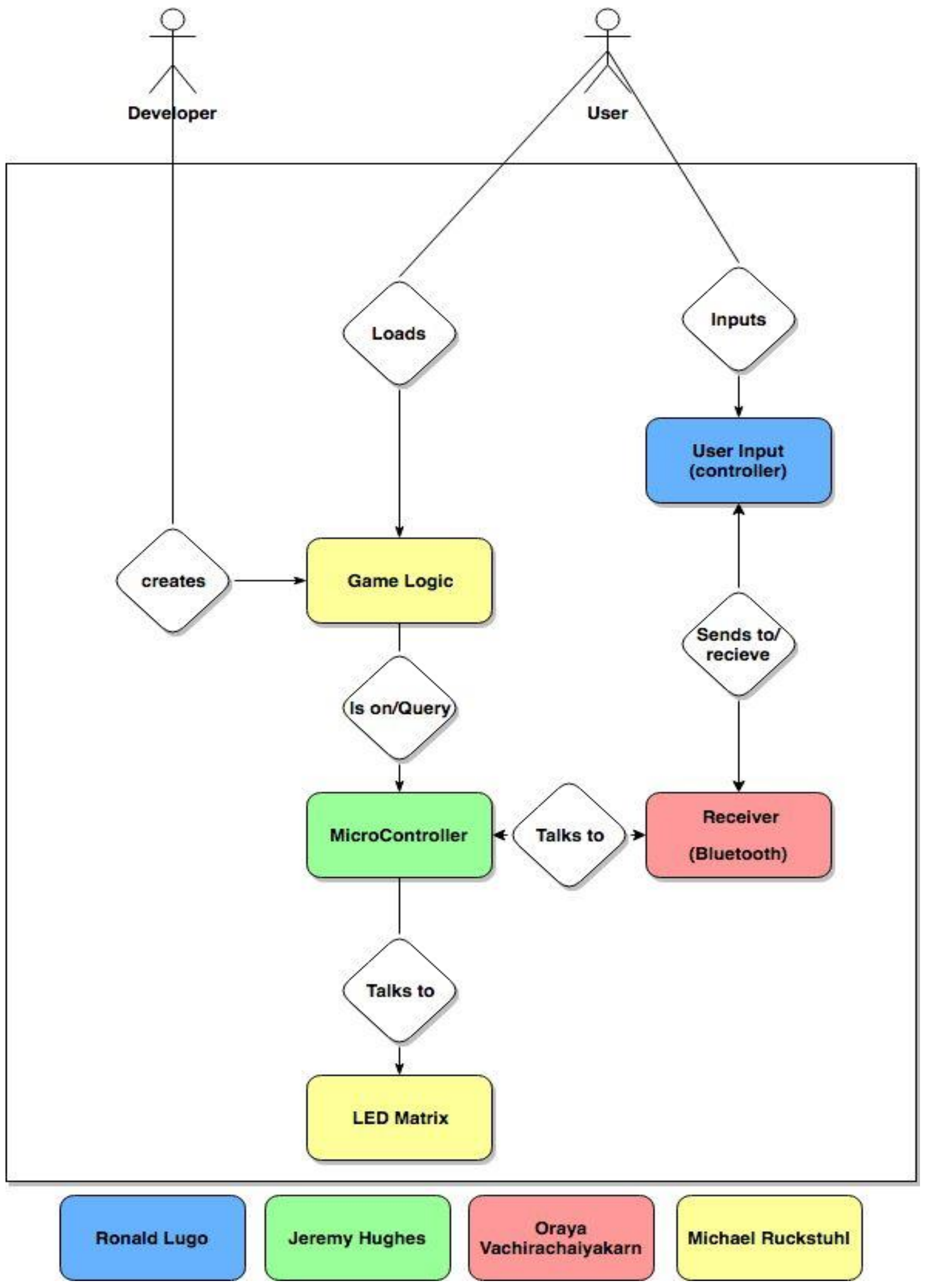


Figure 3 Software Diagram

3. Research Related to Project Definition

The following section details the selection of components for the K.I.T. design and the research procedures that were performed to make these selections. The LED Display, Audio, Game Controllers, Power Supply, and software research is detailed in their respective sections. Furthermore, there are additional researched components that will be part of K.I.T. included in this section.

3.1 LED Display

Designing the LED display is integral to the entire design of this project. The components selected to operate the display greatly affect the components necessary for each other portion of the project. The display will consume the most power and so will be the primary concern when choosing a power supply. The requirements mandate the LED display have a relatively high refresh rate to ensure smooth motion for game display. Finally, the LED display is the main focus for the user and will have a significant effect on the consumers experience. The Display design is broken into two parts, the LED component and the display driver component.

3.1.1 LED Overview

To begin designing the LED display it is necessary to select the proper LED component. It is possible to create an LED display by designing an integrated circuit for each individual LED, however doing so is time consuming and with the individually addressable LEDs that are commercially available it would be ill-advised. Choosing the proper LED component focused around three main ideas; how the LEDs create color, how much power the LEDs consume, and how the LED IC receives information. This section will focus predominantly on three different addressable LEDs available; the WS2812B, SK6812, and WS2813.

3.1.1.1 WS2812B LED

The WS2812B LED manufactured by World-Semi, as shown in Figure 4, is a redesign of one of the first addressable LEDs. Coming in a 5050 package with RGB LED capabilities the WS2812B is capable of meeting basic 8-bit color palette. This means that each LED setting is determined by an 8-bit number. So, each LED has 256 different levels of light intensity. With three LEDs (Red Green and Blue) there are a theoretical 16777216 different color possibilities [17]. However, many of these different color possibilities are indistinguishable as the deviation in light intensity is negligible and so the variety of colors the user can perceive is much less.

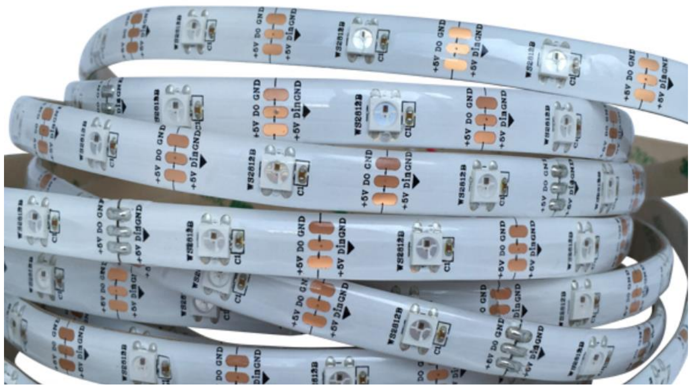


Figure 4 WS2812B LED Strip

The WS2812B operates off a 5-volt source and each LED consumes approximately 0.3 watts. The method of communication the WS2812B LEDs use is often referred to as SPI, however upon further research just referring to the communication as SPI does not fully describe how the LED receives data. Data transmitting speeds of up to 800 Kbps are possible with the WS2812B [53]. For an in-depth understanding of the communication method used refer to the Display Design section below.

3.1.1.2 CS8208 LED

The CS8208 LED of Figure 5, manufactured by WITOP Technology, is similar to the WS2812 addressable LEDs. However, this LED requires a 12-volt signal and does not have LEDs as closely packed as the WS2812 option.

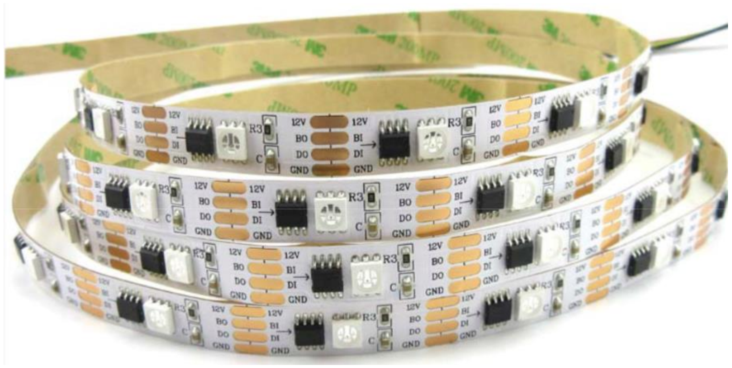


Figure 5 CS8208 LED Strip

The CS8208 has dual signal wires to allow signal break-point continuous transmission. What this means is if one component fails within a strip the components following will continue to receive information. For the CS8208 LEDs mentioned this is not the case as one failing LED will result in a failure of the strip. Similar to the WS2812B, the CS8208 comes in a 5050 package with RGB LEDs capable of meeting basic 8-bit color palette. With three LEDs (Red Green and Blue) there are a theoretical 16777216 different color possibilities [18]. The CS8208 operates off a 12-volt source and each LED consumes approximately 0.24 watts. The method of communication for the CS8208 LEDs is the same as the WS2812 and is often referred to as SPI. For an in-depth understanding of the communication method used refer to the Display Design section below. The CS8208 also has data transmitting speeds of up to 800 Kbps.

3.1.1.3 WS2813 LED

The WS2813 LED is manufactured by World-Semi and is an updated version of the WS2812B as shown in Figure 6. The WS2813 has dual signal wires to allow signal break-point continuous transmission. What this means is if one component fails within a strip the components following will continue to receive information.

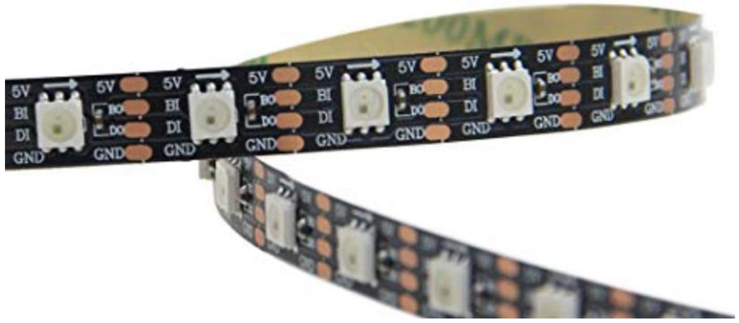


Figure 6 WS2813 LED Strip

For the WS2812B LEDs mentioned this is not the case as one failing LED will result in a failure of the strip. Other than this the WS2813 is the same in that it comes in a 5050 package with RGB LED capable of meeting basic 8-bit color palette [119]. Operates off a 5-volt source and each LED consumes approximately 0.3 watts. The method of communication the WS2813 LEDs use is the same as the WS2812 and SK6812. Also, this LED design offers the same 800 Kbps data transmission speeds.

3.1.1.4 LED Comparison Chart

The comparison between the different characteristics of the LED options reviewed is listed in Table 5. Each LED reviewed exhibits similar technologies and characteristics which was purposely done as their communication methods are well documented. They are compatible with a variety of different commercially available microcontrollers. Though the WS2813 has the added benefit of signal break-point continuous transmission this was not entirely necessity for this project. The K.I.T. is design in such a way that replacing LEDs is not difficult in the case of failure. The CS8208 offers all the same capabilities as the WS2812B with lower power consumption. However, because they operate on a 12-volt source it would require additional power design to compensate as the rest of the subsystems of the K.I.T. operate on 5-volt sources. Finally, the main motivations behind choosing the WS2812B was the documented reliability and the low cost. With the WS2812B coming in at 31% less cost than the SK6812 and 59% lower than the WS2813 going forward with the WS2812B is the best economic decision.

Table 5 LED Characteristics Comparison

Parameter	WS2812B	WS2813	CS8208	Unit
Input Voltage	+3.5 ~ +5.3	+3.5 ~ +5.3	+12	V
Power Consumption	0.3	0.3	0.24	Watt/LED
Working Temperature	-25 ~ +80	-25 ~ +80	-40 ~ +85	C
Data Transmission Speed	800	800	800	Kbps
In-Line Failure Prevention	None	Dual Signal Wire	Dual Signal Wire	US Dollars
LEDs per Meter	60	60	48	-
Cost	$20	$44	$29	Per 60 LEDs

3.1.2 Display Controller Overview

An equally critical component to how the display of K.I.T. performs is the display controller(driver). The display controller is required to operate each and every LED component in the display. This display contains 768 LED components. The controller will be required to update each LED at a relatively high refresh rate in order to achieve smooth motion graphics for gameplay. Understanding how the LEDs will be connected to the controller is one constraint on the controller chosen. For fast refresh rate the design of the LED display is broken down into 8 LED strips of equal length. This means that the controller chosen will need to be capable of transmitting 8 different varying 5-volt square wave pulse signals.

To better understand what each controller will need to be capable of as far as communication to the LEDs it is necessary to define how the LED receives information. The LEDs communicate in a somewhat unique serial peripheral interface. A brief description of this communication is that the first LED in a line of serial connected LEDs latches on to the first 24 bits it receives and transfers every bit afterwards. Bits are transferred every 1250 nanoseconds, which means the period for the signal is 1250 nanoseconds. During that period a high bit is recognized when the input is high for the first 800 nanoseconds and low for the last 450 nanoseconds. A low bit is recognized when the input is high for the first 450 nanoseconds and low for the last 800 nanoseconds. The input is recognized as high if it is at 5 volts DC and low at approximately 0 volts DC.

Certainly, there are tolerances that are acceptable for communicating bits and a more detailed explanation of this communication can be found in the design section. What this means for the controller is that it must be capable of highly accurate pulse width modulation. The controller will need to be capable of alternating the output at a minimum speed of 2.22 MHz. However, the controller must also be capable of this alternating output eight times nearly simultaneously. So now the minimum controller speed is 17.8 MHz. The LED recognizes a reset signal as a low input of 50 microseconds. This essentially sets the tolerance of the accuracy for the controller's internal clock.

If for some reason the controller does not send a high bit within 50 microseconds the display will assume the message is over and the next bit sent will be received by the first LED in the serial connection. Finally, the last main deciding factor for the proper display controller is that it communicates with the game logic controller. The K.I.T. is purposely design to separate display operation and game logic information in an effort to increase the capabilities of game development. The game logic is held on a raspberry pi which transmits display information to the display controller. This is done using a USB interface between the raspberry pi and the display controller. The next three sections will discuss possible display controllers with a focus on speed capabilities, output capabilities, and raspberry pi compatibility.

3.1.2.1 Arduino Mega 2560 R3

The Arduino Mega 2560, shown in Figure 7, is one of Arduino's powerhouse development boards for small scale projects. Research into the multitude of capabilities this development board offers indicates it may be well suited for controlling the display.

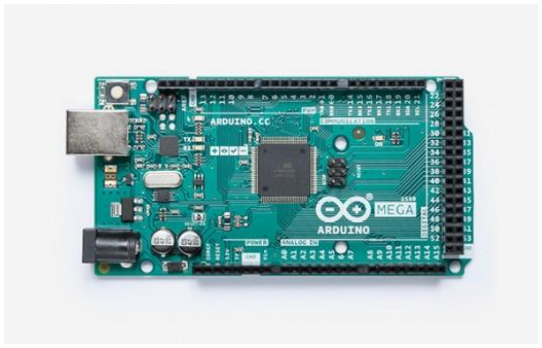


Figure 7 Arduino Mega 2560 R3

In the early stages of research for the K.I.T. this development board was chosen to operate both the display and the game logic. The Arduino Mega 2560 is powered by the ATmega2560 which is capable of 16 million instructions per second or 16 MHz operation. This means it can compute one instruction every 62.5 nanoseconds. The Arduino mega comes equipped with 54 general purpose input/output channels of which 15 are capable of PWM (pulse width modulation) outputs [15].

The Arduino mega also contains four UART communication channels. Which would allow for four USB connections. The 256 KB of flash memory is more than enough to manage the code language used to operate the display. To summarize the Mega comes close to the 17.8 MHz necessary for fast refresh of all LEDs but falls short. However, it is capable of communicating with the raspberry pi via USB and the output signals are 5 volts which is necessary for the LED communication. The Arduino Mega requires an input power of approximately 7-12 volts which means that in order to work with a 5-volt power source it is necessary to implement a voltage amplification circuit. One major benefit of the Arduino Mega is the compatibility with the Fast LED library available from Arduino which is intended for communication with WS2812B style addressable LEDs.

3.1.2.2 Propeller™ P8X32A

The Propeller P8X32A is an 80 MHz microcontroller. Instead of a development board like the Arduino Mega 2560 R3. The Propeller P8X32A is simply the controller unit and a PCB would be constructed with the Propeller in mind. Figure 8 shows the Propeller P8X32A.

Figure 8 Propeller P8X32A

With the high-speed capability, the Propeller is more than capable of handling the speed requirements of the LED display. Equipped with 32 input/output pins this microcontroller also has more than enough channels to handle communicating with the LED display. Beyond the basic control of the LED display though general-purpose input output the Propeller comes equipped with built in VGA capabilities, however, utilizing this would require modifying the output to be compatible with the LED display and is not recommended for the WS2812 LED arrays. Also, there is no existing code libraries for communicating to an LED display using this microcontroller which would require greater software development tasks. Finally, the Propeller does not come equipped with any UART channels which would fail to be compatible with the raspberry pi via USB connection [3].

3.1.2.3 Teensy 3.2 Development Board

The final display controller considered for K.I.T is the Teensy 3.2 development board as shown in Figure 9. Designed by PJRC, the Teensy 3.2 is the perfect development board for operating LED displays with high refresh rates.

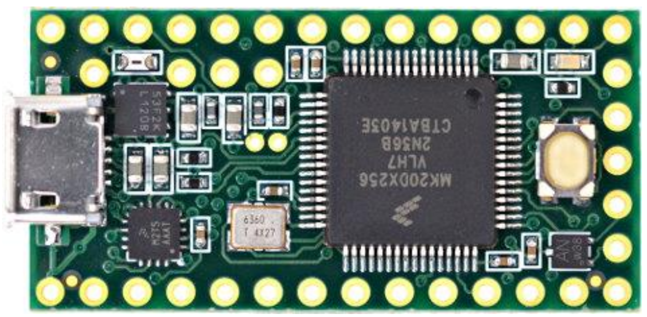


Figure 9 Teensy 3.2 Development Board

The Teensy 3.2 comes equipped with a 32-bit ARM Cortex-M4 72 MHz CPU. This indicates the Teensy 3.2 will be more than capable of updating the LED display. Additionally, the Teensy 3.2 has 16 direct memory access (DMA) channels which can be used to update the LED display faster than a typical general output channel [57]. The Teensy 3.2 also comes equipped with a built-in micro USB port. The Teensy is capable of emulating generic USB devices making it compatible with the Raspberry Pi game logic module. Additionally, the Teensy 3.2 has is compatible with the OCTO2811 led display library reducing the strain on software development for LED display needs. However, the output signal of the DMA channels from the Teensy are 3.3 volts which would require a driver to boost the signal to the necessary 5 volts for the LED display.

3.1.2.4 Display Driver Comparison Chart

The Table 6 below compares the different characteristics of the controllers considered to drive the LED display of the K.I.T. For the superiority in highspeed data transmit capabilities needed for the high refresh rate and the USB compatibility constraints, the Teensy 3.2 is chosen as the K.I.T. display controller. An external power range within the same constrains as the LED strip also makes the Teensy a good candidate. Finally, the Teensy 3.2 rises against competition with the available OCTO2811 LED display library designed for the Teensy 3.2 utilizing the highspeed DMA channels.

Table 6 Display Driver Characteristics Comparison

	Microcontroller		
Specification	Arduino Mega	Propeller	Teensy 3.2
Operating Voltage	5V	3.3 V	5V
Memory	256KB	32KB	256KB
I/O Pins	14	22	20
Dimensions	10.2cm x 5.34cm	4.5cm x 1.8cm	3.6cm x 1.8 cm
USB Support	Yes	No	Yes
DMA Capable	No	No	Yes
Output Voltage	5V	3.3V	3.3V
External Power Range	7-12V	2.7-3.6V	3.3-5.6V
Processor Speed	16MHz	80MHz	72MHz
Price	$38.50	$7.99	$19.80

3.2 Game Controller

The game controller requires a CAD software that can be used to create the pieces that make up the shell of the controller. TinkerCAD was chosen as it is a free, open-source 3D modeling suite. The UCF 3D printers support the STL files which will be produced on TinkerCAD. The exact measurements of each piece must be known before printing as to ensure that the pieces are going to fit together.

3.2.1 Filament types

The filament type is important for each piece of the controller shell. Each filament will give you a different type of feel for your part. The filament types that will be examined are thermoplastic elastomer (TPE), acrylonitrile butadiene styrene (ABS), and polylactic acid (PLA).

3.2.1.1 Thermoplastic Elastomer

TPE's are a rubberlike plastic that are useful in situations where you need a part to be impact resistant, vibration resistant, or flexible. Figure 10 illustrate the flexibility of TPE material.

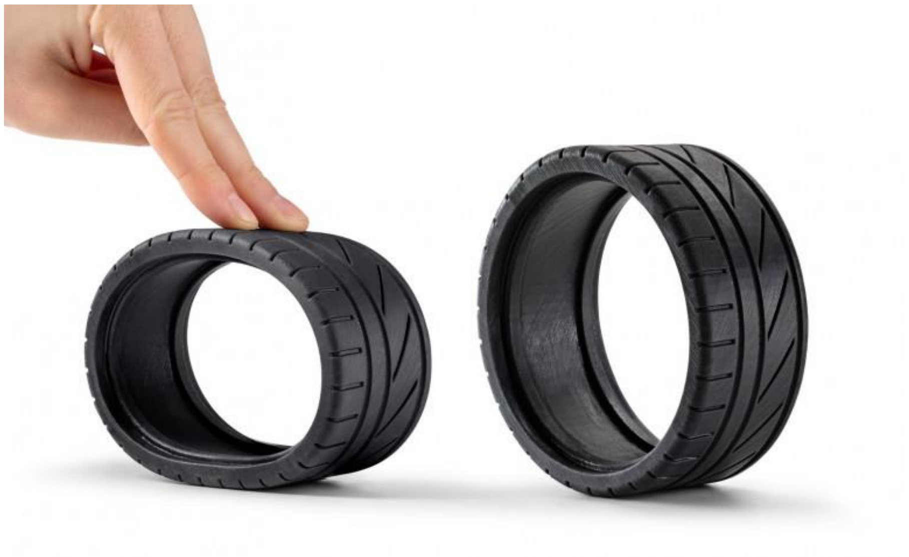


Figure 10 TPE Plastic

They are sometimes hard to print with as there cannot be any gaps or obstacles in the filament path and because the material can stick a little too much to your printing surface. The material should be stored in a low humidity environment to prevent degrading of the material. Depending on the brand, the cost may be inexpensive or more expensive [1].

3.2.1.2 Acrylonitrile Butadiene Styrene vs Polylactic Acid

ABS filament is a strong plastic. It is a popular material among professional businesses as it can be used to make car parts, headgear, and pipe systems. Due to its very high melting point of around 230 degrees Celsius, it is difficulty to print with. If you cool the material while printing, the material tends to warp. Thus, a heating bed is required to obtain the best possible results when printing. While the material is being melted, fumes that are unpleasant and may cause irritation. ABS filament is generally around $20 per kilogram.

PLA filament is a bioplastic that has a shinier and smoother appearance and feel. It is considered safer and more convenient than ABS filament. It has a much lower melting point of around 200 degrees Celsius and can be printed with or without a heating bed. The While undergoing the melting process, the fumes are not unpleasant since the material is sugar based. The cost is about $19 per kilogram [2]. Figure 11 shows the difference between PLA and ABS materials appearances.

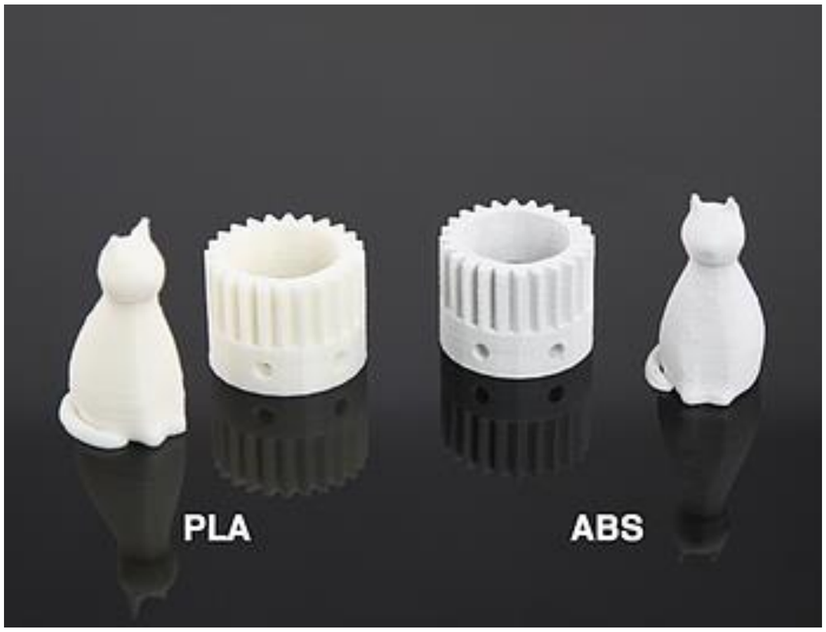


Figure 11 PLA vs ABS Plastic

3.2.1.3 Chosen Filament for Game Controller Parts

After researching each filament type, it was decided that a TPE would be best used to create the buttons as the flexible nature of the material suits button covers very well. For the outer shell of the controller, the PLA filament was chosen as a very strong structural integrity using ABS does not outweigh the convenience of the PLA printing process. The controller will not be subjected to a ton of wear and tear and thus the more convenient material was used.

3.2.2 Possible Microcontrollers and Bluetooth Modules for the Game Controller

This section goes into detail on the possible microcontrollers and Bluetooth modules that could be used for the Game Controller design.

3.2.2.1 Bluefruit EZ-Key

The Bluefruit EZ-Key is a Bluetooth human interface device (HID). It can be used to create a keyboard or controller. It has twelve input pins that can all be connected as buttons. A sample of the Bluefruit EZ-Key is shown in Figure 12.

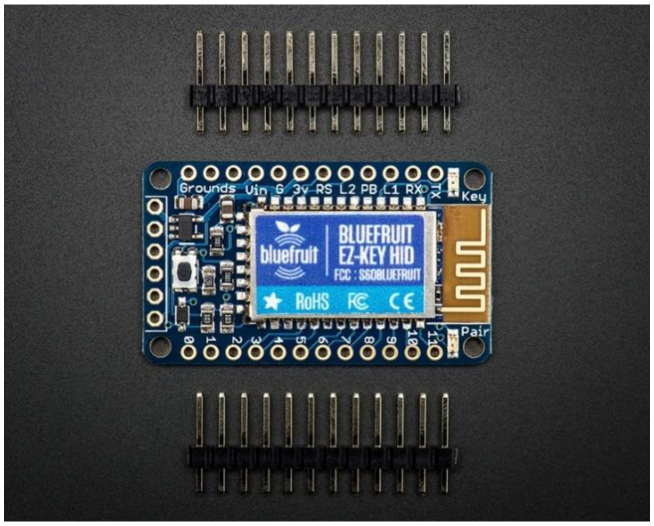


Figure 12 Bluefruit EZ-Key

The Bluefruit makes it possible to connect and program all of the necessary buttons while still having a couple more pins if necessary.

The Bluefruit EZ-Key's main attributes are its simplicity and ease of use. This HID does not require a microcontroller to be connected to it to function. To get the HID up and running, you just have to power it using a 3-16 VDC power supply, connect switches to any of the input pins 0-11 and ground the HID. The HID will then be ready to pair to any device. The default key mapping for each input switch can be found in the Bluefruit EZ-Key user manual. These keys can be reprogrammed using an FTDI or serial console cable [4].

3.2.2.2 Bluefruit LE UART

The Bluefruit LE UART is a low energy Bluetooth module that uses the Nordic UART RX/TX connection profile. It features an entire AT Command set that allows the user to control exactly how the module will behave. It also has the ability to be used as a Generic ATTribute Profile (GATT) which means that it can be configured with GATT commands to be used for data organization, storage and data exchanges between two connected devices [5].

Since the Bluefruit LE UART was designed to pair well with Arduino Microcontrollers, it also uses the Arduino IDE to program the necessary functions that the user would need for their project. Testing the Bluetooth module's capabilities would not require much as there are multiple different Arduino based apps on both Android and IOS that will allow the user to connect to the module and perform functions such as picking a color, showing button presses and streaming data.

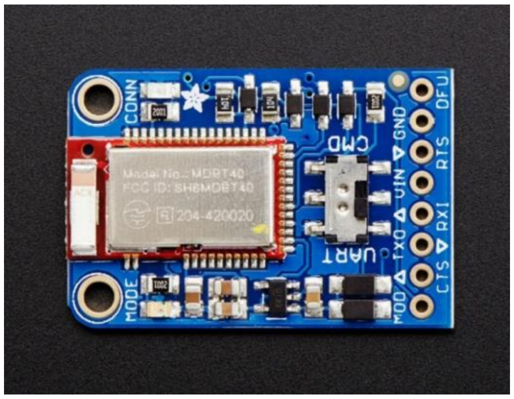


Figure 13 Bluefruit LE UART

Included in figure 13 is displays the Bluefruit LE UART. Figure 14 below displays how to connect an Arduino Uno to the Bluefruit. This enables the Arduino Uno to have wireless communication capabilities. In the case of the KIT game controller design, these two modules could be paired to provide all of the necessary components to drive a wireless/wired game controller.

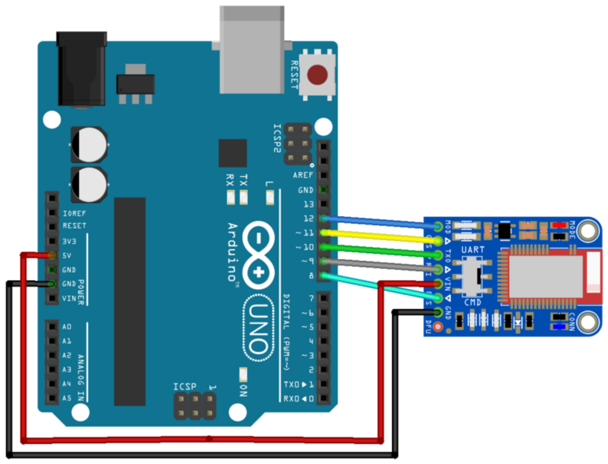


Figure 14 Bluefruit LE UART to Arduino Uno Example [5]

The module can be used as a HID keyboard like the EZ-Key but uses AT commands to send the keyboard data. This module cannot be used to create a controller on its own as it does not input pins to connect switches to. This module must be paired with a microcontroller in order to create a game controller. The microcontroller would be used to connect switches to and the Bluefruit LE would be connected to the microcontroller and used solely for its Bluetooth capabilities [6].

3.2.2.3 Arduino Uno

The Arduino Uno is a microcontroller with fourteen digital input/output pins and six analog inputs. Figure 15 below displays an Arduino Uno.

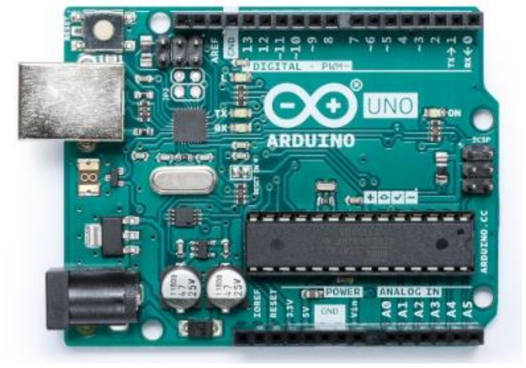


Figure 15 Arduino Uno

It is the most used and documented microcontroller that Arduino offers. It can be powered via USB, AC-to-DC adapter or battery. It can be paired with a Bluetooth HID to create a Bluetooth game controller. The microcontroller has plenty of input pins that can be connected to switches and programmed using the Arduino IDE to fit the requirements of whatever function you may be trying to accomplish. The microcontroller is larger than the previously discussed Bluetooth HIDs. However, its versatility would make it a reliable choice as the base for the game controller.

3.2.2.4 Arduino Nano

The Arduino Nano is a smaller microcontroller that has twenty-two digital input/output pins and eight analog input pins. It uses the same chip as the Arduino Uno but is significantly smaller. It is a breadboard friendly microcontroller making prototyping very easy. Figure 16 shows what the Arduino Nano looks like.

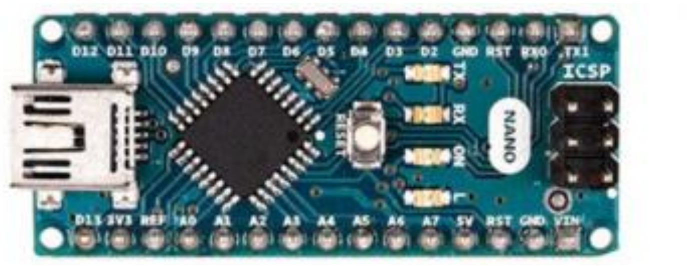


Figure 16 Arduino Nano

The large amount of input pins gives a lot of freedom in game controller design. There are enough pins to allow the user to create and program a complex game controller with many buttons. The microcontroller can be powered via Mini-B USB connection, 6-20V unregulated external power supply via pin 30 or 5V regulated external power supply via pin 27.

The Nano can be connected to a Bluetooth HID to give it Bluetooth capabilities and extend its communication ability. The microcontroller's small size makes it simple to integrate onto a custom PCB for a game controller [9].

3.2.2.5 HC-05

The HC-05 is a basic Bluetooth module that can be connected to a microcontroller to give it Bluetooth capabilities. It is a small module that generally operates at 5V.

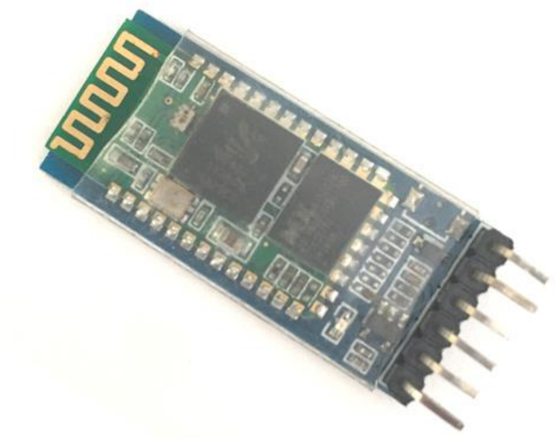


Figure 17 HC-05

The module shown in Figure 17 supports USART and operates using Serial Port Protocol (SPP). This make the module compatible with most microcontrollers including Arduino microcontrollers which are being looked at for the game controller design. The module supports a data mode which allows it to transmit and receive information from other Bluetooth devices and the AT Command mode which allows the user to change the default settings of the module. Choosing one of these two modes is as simple as grounding the key terminal which will put the module in AT Command mode. The module can communicate with computers and phones as well as microcontrollers and can be used in any project to create the ability to receive and transmit data via Bluetooth. In this case, the small size of the module would allow it to be integrated onto a custom PCB that could fit inside a game controller. Figure 18 below displays the HC-05 Module Connections.

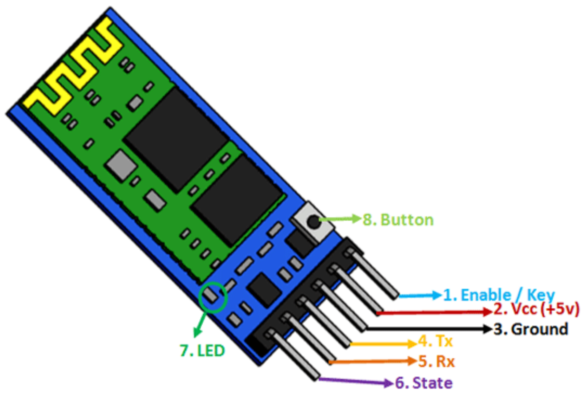


Figure 18 HC-05 Module

3.2.2.6 RN-42

The RN-42, shown in Figure 19, is a Bluetooth HID module that can be connected to a microcontroller to give it Bluetooth capabilities. It is a small, low-power Bluetooth module that operates at around 3.3V.

Figure 19 RN-42

As a HID module it supports multiple profile standards including SPP, DUN, GAP, RFCOMM and L2CAP. The module supports both UART and USB interfaces. The module is compatible with Arduino microcontrollers and would fit on a custom PCB made for a game controller.

3.2.2.7 Arduino Micro

The Arduino Micro is a small microcontroller board with twenty digital input/output pins. Of these pins, twelve can be used as analog inputs and seven as PWM outputs. Figure 20 shows what the Arduino Micro looks like.

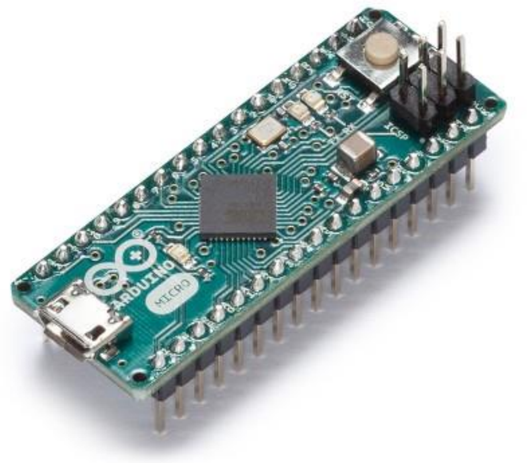


Figure 20 Arduino Micro

The microcontroller has built-in USB communication support which makes it very easy to get up and running. It can be connected directly to a computer and will be recognized and configured as desired. The board can be programmed using the Arduino IDE. The board supports being powered via its micro USB port or by external sources such as a DC power supply or battery with a recommended operating range of 7 to 12V. The Micro is capable of communicating with another microcontroller board of the Arduino family and is compatible with Bluetooth modules. The microcontroller's pinout is shown below in section 4.

The size of the microcontroller and compatibility with Bluetooth modules and other Arduino microcontrollers make the board a viable option to integrate onto a custom-made PCB for a game controller.

3.2.2.8 Comparison of Microcontroller Specifications

For easier comparison between all the microcontrollers discussed in the previous sections, Table 7 shows the key specifications between all the products.

Table 7 Microcontroller Specifications

	Microcontroller			
Specification	Uno	Nano	Micro	EZ-Key
Operating Voltage	5V	5V	5V	5V
Memory	32KB	32KB	32KB	N/A
I/O Pins	14	22	20	12
Dimensions	6.86cm x 5.34cm	4.5cm x 1.8cm	4.8cm x 1.77cm	3.9cm x 2.3cm
USB Support	Yes	Yes	Yes	No
Bluetooth Module Compatible	Yes	Yes	Yes	N/A
External Power Support	Yes	Yes	Yes	Yes
External Power Range	7-12V	7-12V	7-12V	3-16V
Processor Speed	14MHz	16MHz	16MHz	N/A
Price	$22.00	$22.00	$19.80	$19.95
Manufacturer	Arduino	Elegoo	Arduino	Adafruit

The Bluefruit EZ-Key was added to this comparison table as it is capable of being the main component of a game controller. It has the added benefits of it not needing a microcontroller to be paired with to get all of the functionality needed for the game controller design as well as integrated Bluetooth capability. However, the lack of pins and memory when compared to the other microcontrollers ruled this option out.

The Arduino Uno, Nano and Micro all have very similar characteristics with their operating voltages all being at 5V for a regulated power supply and 7-12V for an external power supply. All three microcontrollers have 32KB of memory with additional SRAM storage. Each of these microcontrollers supports USB communication and can be paired with a Bluetooth module to extend its communication abilities. The Arduino Uno has the slowest processor speed at 14Mhz while the Nano and Micro both have a processor speed of 16MHz. The Arduino Uno also has the least number of input/output pins with 14 while the Nano and Micro have 22 and 20 respectively. With all of these specifications including the fact that it is the largest of the microcontrollers, the Arduino Uno was ruled out.

The Arduino Nano and Micro have almost identical specifications. This includes the dimensions of the microcontrollers. The Arduino Nano costs slightly more than the Arduino Micro but comes equipped with two more input/output pins. As the specifications are practically identical, either microcontroller could be chosen and used to accomplish the desired goal of designing a game controller. However, the extra pins on the Arduino Nano would give slightly more flexibility in the design process. Thus, the Arduino Nano was chosen as the basis for the game controller's design.

3.2.2.9 Comparison of Bluetooth Module Specifications

Again, the Table 8 shows the comparison of specifications between the Bluetooth Modules discussed in the previous sections.

Table 8 Bluetooth Module

	Bluetooth Module			
Specification	EZ-Key	LE-UART	HC-05	RN-42
Operating Voltage	5V	5V	5V	3.3V
Operating Current	25mA	13.5mA	30mA	40mA
Needs Microcontroller	No	Yes	Yes	Yes
Dimensions	3.9cm x 2.3cm	3.18cm x 2.13cm	3.75cm x 1.61cm	2.58cm x 1.34cm
Range	10m	10m	100m	18.3m
Price	$19.95	$17.50	$7.69	$18.95
Manufacturer	Adafruit	Adafruit	DSD Tech	Microchip

As stated in the previous section, the Bluefruit EZ-Key was ruled out due to its lack of versatility. The rest of the discussed modules are very different from each other. All are different sizes although all of them are small enough to be integrated onto a custom PCB. The LE-UART's main drawback is its 10m transmit/receive range which is the maximum range as it generally transmits/receives at 6m. Although it is the lowest power consuming of the modules, it was ruled out due to its poor range capabilities.

The HC-05 and RN-42 are both capable of performing the required task needed for a Bluetooth module in this game controller design. However, the HC-05 has a sizeable advantage in range capabilities and cost. The low cost of the HC-05 would allow for two modules to be bought for the same price as one RN-42 to provide the necessary backup in case the hardware malfunctions. Thus, the HC-05 was chosen as the module for the game controller. This is turn eliminates any issues that may come about if two incompatible Bluetooth modules were chose for the design as the module chosen will support the same profile protocols as the Bluetooth module used for the LED game board is the same module type.

3.2.3 LCD Display Options

A Liquid Crystal Display, LCD, will be used to display various things including, but not limited to, the status of Bluetooth connection between handheld controllers and K.I.T., debugging, troubleshoot, internal temperature, brightness level, volume, and current game that is being played. A Python programming language can be used to program the LCD display that is to be connected to the Raspberry Pi microcontroller.

There are several options of selecting the LCD screen for each having different sizes and functions. The following sections discuss the details of the LCD Display options and compare them to obtain the most suitable LCD display.

3.2.3.1 SunFounder LCD 2004

The first option for KIT is LCD2004 from the company SunFounder as shown in Figure 21. It is compatible with Arduino Uno, Mega2560, and Raspberry Pi. The display is capable of displaying 20 characters each line for 4 lines using white characters on a blue background. The display dimension is 71 millimeters by 21 millimeters while the entire module dimension is 97 millimeters by 59 millimeters by 12 millimeters. The LCD display only weights 1.76 ounce, great for KIT as we want the device to be as lightweight as possible. The LCD2004 operates on a 5-volt DC input and has an adjustable 50k potentiometer on the side to adjust the contrast [22]. There are total of 16 pin headers where pins D0 through D7 are used to read and write data and pins A through K to control the LCD backlight.

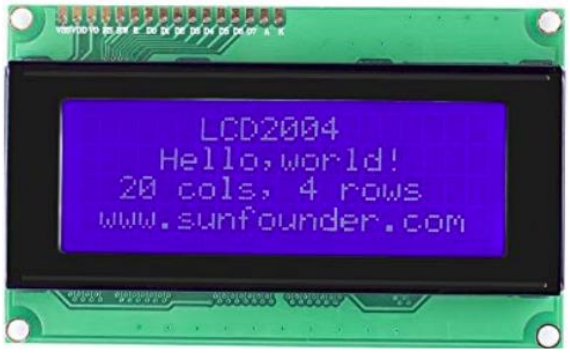


Figure 21 SunFounder LCD2004 Module

The gaming microcontroller Raspberry Pi can be connected to the LCD2004 display by connecting the pins together via a breadboard or a PCB. This LCD display needs an external potentiometer to control the display.

3.2.3.2 Adafruit Pi Plate LCD 16x2 Screen

Another option for the LCD screen is the new product from Adafruit. The product includes the Adafruit Pi Plate that is compatible with the included blue and white 16 column by 2 rows character LCD. The LCD screen displays a white text on a blue background. It has a single LED backlight that is dimmable using a resistor [83]. The dimension of the LCD screen itself is 24 millimeters by 69 millimeters. This Pi plate has the ability to control a 16x2 character LCD, up to three backlight pins, and five keypad pins using two I2C pins on the Raspberry Pi [84]. The Figure below shows the LCD screen attached to the Arduino Pi plate with the four directional buttons and a select button on the right side of the Pi plate. It also has a contrast adjusting potentiometer to adjust the contrast of the LCD screen [84]. The kit also comes with the PCB that controls the LCD screen and the Pi plate, making it easier to assemble. Figure 22 shows what the LCD kit looks like when it is operating.

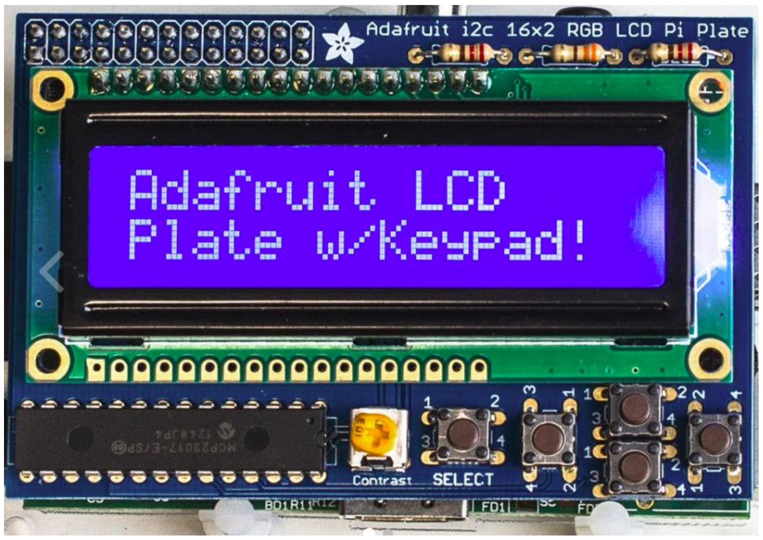


Figure 22 Adafruit LCD Plate

This Adafruit Pi plate is compatible with Raspberry Pi, and can be programmed using Python coding language. It operates at 5 volts DC for both logic operating and for the backlight [82]. The operation temperature is between -10 degree Celsius and 60 degree Celsius [82]. Its maximum operating current rating is roughly 2.5 milliamp.

3.2.3.3 LCD Display Selection

After all the research, Table 9 shows the comparison between the Adafruit LCD and SunFounder LCD2004. They are both similar in features and operation functions. The main difference is the size between the two, the SunFounder LCD2004 is almost two times bigger than the one from Adafruit. For the convenience of assembling, the group has decided to go with the Adafruit LCD since it comes with all the premade PCB board and selection buttons.

Table 9 LCD Screen Comparison

Features	SunFounder LCD2004	Adafruit LCD
Operating Voltage (V)	5	5
Size (mm x mm)	59 x 97	24 x 69
Display (column x row)	20 x 4	16 x 2
Operating Current (mA)	0.25	2.5
Price ($)	7.99	19.95

3.3 Software Research

During the research portion of the project, one design question that needed to be answered was what programming language would the project use? Most of the project would consist of microcontrollers talking to one another. For most micro controllers, the programming language would be limited by whatever language the hardware would accept. This could be anything from assembly, C, or to any more high-level programming language. Most of the micro controllers we were choosing for the project were Arduinos. Conveniently Arduino allows users to program in C or C++ making things simple [89]. However, other micro controllers allow the use of more complex object-oriented languages like Python.

The language specifications generally depend on the hardware being used for the project. Most micro controllers that are used on this project will most likely be programmed in C simply because they are Arduino. For the game logic, a separate consideration was used. In order to give programmers greater flexibility and to increase the general appeal of the K.I.T., we wanted to allow the game logic to be programmable in a more accessible language like Python. This pushed our choice of hardware for the game logic toward the Raspberry Pi. Since the Raspberry Pi uses Python natively, it was an ideal choice.

Using Python would allow future programmers to be less restricted by C or assembly level code. Python would also allow future programmers to be more object orient in their code and allow them to use any of the many libraries written for Python for their projects. Other hardware considerations also contributed toward the decision as well. Another consideration was that the main programmer for the project had knowledge of Python. For these reasons, Python and the Raspberry Pi was chosen for handling the game logic.

3.3.1 Software Microcontroller

For the project's design, it was decided that there needed to be a dedicated microcontroller to handle the game logic. This micro controller would then communicate with the other parts of the K.I.T. to fulfill whatever program it had on it. The microcontroller we wanted was something that could be easy to program and would fit within the hardware spec. For the software microcontroller, several options were considered to run the main programs of the project. The main contenders for the microcontroller were the Raspberry Pi, Arduino Uno, and the Arduino Mega. The hardware specifications of the Arduino Uno, and the Arduino Mega were mentioned in section 3.4.2.3 and 3.1.2.1 respectively. Instead of repeating those specs, this section will focus on the software aspects of those microcontrollers.

3.3.2 Raspberry Pi 2

The Raspberry Pi 2 is a microcontroller with the low power high performance Cortex-A7 MPcore ARM processor [13]. The processor actually consists of four Cortex-A7 processors acting as one and runs at around 1.5 GHz collectively. Each individual processor runs at 900MHz. It also has a floating point unit and an L2 cache making it very powerful. The Raspberry pi 2 has an overall memory of 1 GB. The board also has several ports including 4 USB 2.0 ports, an HDMI port, a 3.5mm phone jack, and an Ethernet port [90]. The Raspberry Pi 2 also has 27 GPIO pins to use in the project. Figure 23 shows the GPIO pins of the Raspberry pi 2.

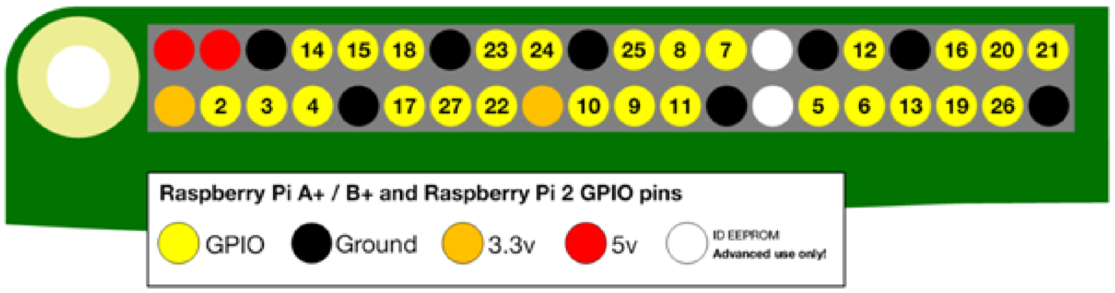


Figure 23 Raspberry Pi GPIO Pins [14]

The Raspberry Pi 2 was attractive from a software perspective since it was able to run Python programs and has a native OS. Having an OS makes programming for the Raspberry Pi 2 simpler since testing can be done in an OS environment instead of simply monitoring pin outputs. The Pi's ability to program in Python allows greater complexity with the code for the games. Allowing Python programming will also ensure that future programmers will be more inclined to program for the K.I.T. on their own projects. Python is a popular programming language with many users. The Raspberry Pi also has many tutorials and documentation to make integrating this board into our project easier. A drawback was the significant price increase at 39.95. This higher price could cut into the budget for other micro controllers and was considered in comparison evaluation.

3.3.3 Arduino Uno

The Arduino Uno uses the ATmega328P from Microchip as the main processor. The software specifications of the Arduino Uno include 32KB of Flash memory and a clock speed of 16MHz (0.5 KB used by the bootloader) [9]. The Arduino Uno uses C or C++ to program using the Arduino IDE.

The Arduino Uno was considered for the K.I.T. since the hardware specs matched what the project needed, and it was already being considered for other parts of the project also. The price was also very reasonable at 22.50, keeping with our budget. The main draws for the Arduino Uno for controlling the game logic was the simplicity of the board, convenience, price, and the simplicity of the language. As the project moved on, several issues with this arose. We wanted to be able to store multiple games at once and be able to access all of them without having to reload each one. We also wanted to be able to widen the scope of the games we were programming and give future programmers the ability to write larger more complex programs. The Arduino's small flash size and its use of C/C++ were drawbacks for the project. The simple hardware made it an attractive choice, but we looked for other options.

3.3.4 Arduino Mega

The Arduino Mega uses the Microchip ATmega328P. With this board, user get 256 KB of Flash memory (8 KB used by bootloader), and a 16 MHz clock speed [15]. The Arduino Mega also uses Arduino's C/C++ language to program in.

The Arduino Mega was also considered for the project to overcome shortfalls with the Arduino Uno. The price point for the Mega was 38.95, still making it cheaper than the Raspberry Pi 2. The Arduino Mega is marketed for more complex projects and has double the memory of the Arduino Uno. The memory increase gave the project more options over the Arduino Uno, but several things held the Arduino Mega back. The clock being only 16Mhz was considered a liability. We wanted to be able to refresh the screen at a decent rate and felt that the clock speed might turn into a bottleneck in the pipeline that would slow the process of refreshing the screen down.

The language native to the Arduino was another sticking point. C/C++ would be easy to code in but could possibly limit the complexity and scope of the games that could be programmed. The Arduino Mega's simple design was attractive, but there were several potential issues with it that could have affected the project later on down the line.

3.3.5 Choice Comparison

The three contenders for the microprocessor that would run the game logic were the Arduino Uno, Arduino Mega, and the Raspberry Pi 2. Each had different drawbacks to consider as shown in Table 10. The Arduinos had the advantage of low power consumption and a much more reasonable price. The Arduino Uno had the drawback of a small memory size. This was solved with the Arduino Mega, but another issue was the slow clock speed. The Raspberry Pi 2 has the advantages of memory size, clock speed, and programming language. The Raspberry Pi also had the most power consumption and cost of all the microprocessors considered. Overall, we chose the Raspberry Pi 2 to be the game logic microprocessor for the flexibility it offered.

Table 10 Software Microprocessor Comparison

	Arduino Uno	Arduino Mega	Raspberry Pi 2
Clock	16 MHz	15MHz	1.5 GHz
Memory	32 KB	256 KB	1 GB
Language	C/C++	C/C++	Python
Price ($)	22.00	38.50	39.95
Power	45 mA (0.225 W)	200 mA (1 W)	750 mA (3.75W)

3.4 Power

Power consumption of each component inside K.I.T. will have to be taken under consideration to figure out the final power required for operation. The followed sections discuss the details of the power consumption of each component selected for the project along with the comparison between the power supplies that are being considered. Types and sizes of wires are also important for selection purposes and are discussed in the following sections as well.

3.4.1 Power Consumption of Components

There are several components in KIT that require the use of power. The sections below briefly discuss the power consumption of each component including the 5 volts DC LED strips, microcontrollers, development board, case fans, and LCD screen display.

3.4.1.1 LED Power Consumption

When it comes to powering LED strips, it is crucial to supply enough power to all LEDs in order to avoid the dimming of LEDs towards the end of each strip. However, special care must be taken to not burn out the LEDs by applying too much supplied power. LED strips utilize low voltage DC [16], since KIT's design has a total of 768 LEDs, it will get its power from a power outlet rather than from a battery source. For this reason, an AC to DC converter power supply device is needed to supply each strip with the required power.

3.4.1.1.1 5 Volt DC RGB LED Strips

The chosen 5-volt DC LED strips, WS2812B, according to its datasheet, consume about 0.3 Watt per LED [17]; the design consists of 768 LEDs total with 24 strips of 32 LEDs each. The power consumption for each strip and the total power consumption can be calculated as follow: $P_{strip} = 32 \times 0.3 = 9.6\,Watt$ and $P_{total} = 9.6 \times 24 = 230.4\,Watt$. The current is needed to be calculated in order to choose the rating of an AC to DC power supply adapter. Using a basic power equation, the current that will travel through each strip is $I = \frac{P_{strip}}{V} = \frac{9.6}{5} = 1.92\,A$. To calculate the total current that will be needed from the power supply, 1.92 amp is multiplied by 32: $1.92 \times 32 =$

$46.08\,A_{total}$. This means the power supply will have to be able to supply at least 230.4 Watt or, in other word, 46.08 Amps. Due to the high number of LEDs that KIT is designed to have and the low 5-volt DC that the LED strips require, a power supply that has the capability of supplying that much wattage and current will be hard to find and expensive.

3.4.1.1.2 12 Volt DC RGB LED Strips

Another type of LED strips that was being considered is the 12-volt DC LED strip, SMD5050. Each LED consumes 0.24 Watt [18] or the total power consumption of $768 \times 0.24 = 184.32\,Watt_{total}$, about a quarter less than the consumption of the 5-volt LED. The current needed to light up 768 LEDs can be calculated as: $I = \frac{P_{total}}{V_{dc}} = \frac{184.32}{12} = 15.36\,A_{total}$. It can be seen that the current for 12-volt LEDs is only a third of the total current needed for the 5-volt LEDs. In this case, an AC to DC converter will have to supply only 200 Watt, 12 volts, or 16.5 Amp instead of 300-Watt, 5-volt, 60 Amp in the case of 5-volt LED strips. Although this is a great option since the power needed is very low, it might not be the best option for K.I.T. as all the other components within KIT requires 5 volts DC to operate.

3.4.1.2 Raspberry Pi 2 Power Consumption

The Raspberry Pi 2 operates at 5 volts. Depending on the peripherals that are attached and drawing power from the Raspberry Pi 2, the power consumption can be different. The GPIO pins can draw 50mA total. The HDMI port uses 50mA. The Raspberry Pi 2 has a minimum power consumption of around 220 mA (1.1W) when idle and a maximum of around 820 mA (4.1 W) under stress. On average the Raspberry Pi should run at 750 mA (3.75 W) on average [19]. The Raspberry Pi itself can be powered from a micro USB or through pins.

3.4.1.3 Arduino Nano Power Consumption

Arduino Nano microcontroller is similar to the Uno microcontroller, it operates at the voltage of 5 volts. Each pin can receive or provide a maximum of 40 milliamps per input/output pins; the microcontroller consists of 22 digital input/output pins. According to the datasheet, the total power consumption of the Nano microcontroller is 19 milliamps or 0.095 Watt. To power the Nano, a regulated external power supply of 5 volts DC will be needed and will be connected to pin 27 [9].

3.4.1.4 Teensy 3.2 Development Board Power Consumption

Teensy 3.2 is a small development board that is a 32-bit ARM Cortex-M4 platform. It is a USB based microcontroller that has a regulator chip that is capable of taking the input voltage of up to 10 volts [20], a regulated external power supply of 5 volt will be perfect for this development board to function properly. The board typically operates at the voltage of 3.3 volts at medium frequency. At the high frequency of 120 MHz, the Teensy

3.2 consumes about 45 milliamps which is equivalent to 0.1485 Watts, which is still very small for the power supply to handle.

3.4.1.5 HC-05 Power Consumption

HC-05 module is a Bluetooth Serial Port Protocol module used for a wireless serial connection. It operates at the voltage range of 4 volt to 6 volts. An input voltage of 3.3 volts to 5 volts will be perfect to power up this Bluetooth module. The device typically operates at the current of 30 milliamps. Pin 12 is the 3.3 VCC integrated 3.3 volts supply with the built-in linear regulator output of 3.15 volts to 3.3 volts [21].

3.4.1.6 SunFounder LCD2004 Module Power Consumption

LCD2004 is a decent size LCD display of 20 columns by 4 rows. It can be programmed to display letters, symbols, and numbers. It operates at 5 volts DC input voltage at pin VDD [22], a power supply of 5 volts DC output will be perfect to power up the LCD screen. The typical supply current is a very low value of 0.1 milliamp [23]. The maximum supply current is also quite low, at 0.25 milliamp or 0.00125 Watt.

3.4.1.7 GDSTIME GDT8010S12B Cooling Fans Power Consumption

The case fan GDT8010S12B from GDSTIME is a decent size cooling fan; several fans will be used to keep the inside temperature of KIT under control. It operates at a rated voltage of 5 volts DC at various speeds of 1600 RPM, 2200 RPM, and 2800 RPM depending on the setting. Therefore, the power consumption at each speed will be different for each fan. At the speed of 1600 RPM, the current is rated at 0.12 amps [24], resulting in the power consumption of 0.6 Watt using the equation $P = I \times V$. When the speed is set to 2200 RPM, the fan will use 0.15 amps [24] which is equivalent to 0.75 Watt. At the highest speed of 2800 RPM, the case fan consumes 0.18 amps [24] or about 0.9 Watt. If the highest speed of 2800 RPM is to be used for six case fans, the total power consumption will be 5.4 Watts.

3.4.1.8 KIT Total Power Consumption

After the research is done for all of the components that will be included inside K.I.T., it can be concluded that all of the components within KIT operate at 5 volts DC. The main component that consumes the most power is the LEDs, consuming up to 230.4 Watt at its full brightness. The Table 11 shows the power consumption of each component along with the total power consumption consumed by KIT when everything is operating at its full speed. The case fans and Raspberry Pi 2 also consume a good amount of power.

The total power consumption of KIT is roughly 240.6 Watts. A power supply of 250 Watts will be enough to power up KIT at its highest performance. However, a power supply of 300 Watts will be used to ensure that KIT gets more than enough power and will operate without any issue.

Table 11 Total Power Consumption

Component	Power Consumption (Watts) Each	Number of Components	Power Consumption (Watts) Total
5 Volt DC RGB LED	0.3	768	230.4
Raspberry Pi 2	4.1	1	4.1
Arduino Nano	0.095	1	0.095
Teensy 3.2	0.1485	1	0.1485
HC-05	0.15	3	0.45
Adafruit LCD	0.0125	1	0.0125
GDT8010S12B	0.9	6	5.4
Total Power Consumption (Watts)	**240.606**		

3.4.2 Power Supply and Battery

KIT is designed to be powered by the common 110-220 volts alternating current outlet in the United States. An AC to DC converter power supply will be needed since the LEDs require the input of 5 volts direct current and the microcontrollers will also require a DC input voltage. The chosen manufacturer of power supplies is MEAN WELL as they are globally recognized for their safety standards. The types and ratings of power supply being considered for each component of KIT are discussed in the sections below.

3.4.2.1 Power Supply RSP-320-5

An RSP-320-5 is a 320-Watt single output power supply with a power factor correction function. This power supply works with all AC input at full range as it is universal. This power supply converts an AC input voltage to 5-volt DC output voltage at a rated current of 0 ~ 60 amps. Its rated power is 300 watts. The maximum ripple voltage and noise is rated to be 150 millivolt-peak-to-peak. Also, the output voltage can be adjusted between 4.5 and 5.5 volts with the tolerance of ± 2.0%. It has a built-in active power factor correction function of a very high-power factor greater than 0.95. The power efficiency for this model is as high as 83 percent. It has 3 ports for output connections while KIT LEDs will need at least 24 connections. In this case, 8 wires will be connected to each port of the power supply.

This device also has all the protections against short circuit, overload, over voltage, and over temperature with its built-in DC fan that forces air cooling. The overload protection is up to 105 ~ 135% the rated output power that shuts down the output voltage and automatically recovers after the fault condition is cleared. The over voltage protection is of 5.75 ~ 6.75 volts. As per over temperature protection, the device will shut down the output voltage and re-power on to recover. This device working temperature is around -30 ~ +70°Celsius. Its built-in DC fan comes with a fan speed control function.

Furthermore, this device has been approved by multiple safety standards including UL60950-1, TUV EN60950-1, EAC TP TC 004, and CCC GB4943.1, with compliance to EN55032 (CISPR32) Class B and GB9254 Class B [25]. This model costs roughly $45, a great option for being within the budget. Its dimension is 215-millimeter-long by 115-millimeter-wide by 30 millimeters tall, this allows it to fit inside KIT with room to spare.

Figure 24 shows what the RSP-320-5 looks like from the outside. It can be seen that it is a small and tidy power supply with a built-in fan in the corner. It has 3 terminals for positive voltage and 3 terminals for return voltage. It also has an LED light indicator to indicate whether the power supply is turned on or not.

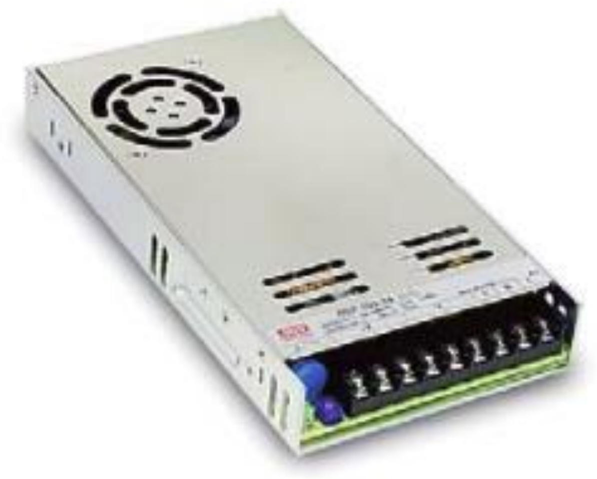


Figure 24 RSP-320-5

3.4.2.2 Power Supply HRP-300-5

An HRP-300-5, shown in Figure 25, is a 300-Watt single output power supply with power factor correction function. It has the ability to convert all alternating current input at full range to a digital current output of 5 volts with the current ranging from 0 ~ 60 amp. The output ports of this power supply only have two ports, KIT will have 24 connections from a power supply, which means 12 wires will be connected to each port. The ripple and noise are only 90 millivolts (peak-to-peak) while the output voltage is adjustable within the range of 4.3 to 5.8 volts. This device has a very small hold up time; at 16 millisecond/230 VAC and 16 millisecond/115 VAC at full load. Its built-in active power factor correction is rated higher than 0.95 and the efficiency of the power supply is as high as 89 percent. HRP-300-5 also has the ability to withstand a surge input of 300 VAC for up to 5 seconds.

The protections are similar to the ones of RSP-320 model, including short circuit, overload, over voltage, and over temperature protections. The overload protection works for 105 ~ 135% of the rated output power and the over voltage protection is of 6 ~ 7 volts. For the over temperature protection, the device shuts down output voltage when the over temperature is detected, then it automatically recovers after temperature has decreased. Its working temperature is rated between -40 ~ +70°C. This device comes with a built-in constant current limiting circuit, great for usage with LEDs since LEDs are very sensitive to current. It has a built-in cooling fan that has an ON-OFF control and a built-in DC OK signal. The fan control is at a load of 35 ± 15% or RTH2 ≥ 50°C. There is also a remote sense function and it comes with a 5 years warranty.

MEAN WELL HRP-300 series have been approved by multiple safety standards including, but not limited to, UL60950-1, TUV EN60950-1, EAC TP TC 004 and EN55032 (CISPR32) Class B [26]. This model costs $80, almost doubled the cost of RSP-320, but it comes with many more functions that can be more useful and suitable for KIT. Its dimension is 199-millimeter-long by 105-millimeter-wide by 41 millimeters tall; it is not too big and should fit inside KIT nicely. With the built-in fan, KIT could become noisy when it is powered on.

Figure 25 HRP-300-5

3.4.2.3 Power Supply HRPG-300-5

An HRPG-300-5 is also a 300-Watt Single Output with a built-in power factor correction function. It is adaptable to a full range of alternating current input which is converted to a 5-volt direct current output within the current range of 0 ~ 60 amps. The ripple and noise are tested to be within 90 millivolt-peak-to-peak. The output voltage can be adjusted between 4.3 volt to 5.8 volt with the voltage tolerance being ±2.0%. Its built-in active power factor correction function has a high-power factor of at least 0.95 with its high-power efficiency of up to 82 percent. Similar to HRP-300 series, this device can withstand a 300 VAC surge input for 5 seconds. The power supply has two ports connections for the outputs.

This device, as shown in Figure 26, has protections against short circuit, overload, over voltage, and over temperature. The overload protection works for 105 ~ 135% of the rated output power by limiting the constant current and automatically recovers once the fault condition is cleared. The over voltage protection is also of 6 ~ 7 volts. For the over temperature protection, the device shuts down output voltage when the over temperature is detected, then it automatically recovers after temperature has decreased. Its built-in constant current limiting circuit makes this model suitable for powering LEDs for its high current sensitivity. The working temperature while the power supply is ON is ranged between -40 to +70°C.

It also has a built-in cooling fan that has an ON-OFF control at the load of 35 ± 15% or RTH2 ≥ 50°C and a built-in DC OK signal that turns the PSU on at 3.3 ~ 5.6 volts and off at 0 ~ 1 volt. Furthermore, HRPF-300 series has a built-in remote ON-OFF control; power on at 4~10 volt or open and power off and 0 ~ 0.8 volt or short; although this is probably not going to be needed as KIT will have an external ON-OFF switch control. This device has a standby 5 volt at 0.3 amp with the tolerance of ±5% and the maximum ripple voltage of 50 millivolt-peak-to-peak. Similar to the other model, it also has a built-in remote sense function and comes with a 5 years warranty. It also has no load power consumption of less than 0.5 watt.

Similar to HRP-300 series, MEAN WELL HRPG-300 series have been approved by multiple safety standards including, but not limited to, UL60950-1, TUV EN60950-1, EAC TP TC 004 and EN55032 (CISPR32) Class B [27]. The cost of this device is $120, very expensive and is tremendously higher than the estimated budget for a power supply. This series is more expensive the HRP-300 series with more functions that KIT does not necessarily need. It shares the same dimension as the other series: 199-millimeter-long by 105-millimeter-wide by 41 millimeters tall.

Figure 26 HRPG-300-5

3.4.2.4 Power Supplies Comparison

For a side-by-side comparison, Table 12 below can be used to compare the specifications between the three types of power supply. The chosen power supply is MEANWELL RSP-320-5. There is no big difference between the technical data between all three of the power supplies while the RSP-320-5 is the cheapest at only $45. It is also smaller than the other two power supplies, making it ideal for fitting inside K.I.T. The only downside is its high ripple voltage of 150 millivolts peak to peak compared to the other two models of 90 millivolts peak to peak. Another reason that the RSP-320-5 is chosen over the other two is because it contains three ports output instead of only two, giving the overall project more flexibility when it comes to wiring.

Table 12 Power Supply Comparison

	RSP-320-5	**HRP-300-5**	**HRPG-300-5**
Price	$45	$80	$120
Dimension mmxmmxmm	215x115x30	199x125x40.1	199x125x40.1
Universal AC input / Full range	Yes	Yes	Yes
Built-in active PFC function	PF>0.95	PF>0.95	PF>0.95
Built-in cooling fan	Yes, with fan speed control function	Yes, with ON-OFF control	Yes, with ON-OFF control
Built-in DC OK signal	N/A	Yes	Yes
Standby 5V@0.3A	N/A	N/A	Yes
Built-in remote sense function	N/A	Yes	Yes
Max Ripple and Noise Voltage	150 mVpp	90 mVpp	90 mVpp
Voltage Adjustment Range	4.5 ~ 5.5 V	4.3 ~ 5.8 V	4.3 ~ 5.8 V
Number of ports	3	2	2
Typical Efficiency	83%	82%	82%
Working Temperature	-30 ~ +70 °C	-40 ~ +70 °C	-40 ~ +70 °C
Storage Temperature	-40 ~ +85 °C	-40 ~ +85 °C	-40 ~ +85 °C

3.4.3 Wires

The type and size of wires used to wire each component together are important especially when the current between the components is high. This is especially true for the wirings of power supplies as the input AC voltage from a wall outlet of 120 volts is very high and the LEDs consume high wattage, leading to a high current needed to feed the LEDs. The size and type of wires used to wire between microcontrollers and other components will not matter as much since the currents needed are very low in ampere. The sections below discuss the type and AWG size of the wires chosen for the project.

3.4.3.1 Power Supply AC Input Connection

The RSP-320-5 power supply from MEANWELL is to be used to supply power to the LEDs by converting the 120 volts input alternating current power to 5 volts output digital current power that will feed K.I.T. The power supply typically takes the input current of 2.7 amps [25], according to Table 13, an 18 Gauge AWG wire will work just fine. However, to be on the safe side, K.I.T. will utilize the 16 Gauge AWG wire, as highlighted in the table below, to connect the power supply to the AC outlet power source, in the case that there is any inrush current. Table 13 shows parts of the circuit-conductor ampacity and voltage rating for jacketed cords in general-use cord sets and power-supply cords per UL guidance [28].

Table 13 General AWG Rating for Jacketed Cords

Gauge Size Range (AWG)	Ampacity of Current-Carrying Conductors (amperes)	Voltage Rating of Cord (Volts AC)
18	7	300
16	10	300
14	15	300
12	20	300
10	25	300

The three-prong power cable will be used as the RSP-320-5 has the ground, neutral, and live connection ports. The chosen cable is from the company *Cable Matters*, it is a heavy duty, though flexible, 16 Gauge AWG 3 prong power cord made with copper conductors. Each wire is insulated individually into white, black and green. The green wire is for protective ground, white is for neutral and black is for live or active single-phase wire [29]. This particular cord is rated for 13 amps to support common power requirements. The outside cover is made of a flexible thermoplastic PVC jacket that is excellent for heat resistance [30]. Heavier gauge cable is more reliable and is safer than the 18 AWG cable. This power cable is rugged and safe with a standard NEMA 15-5-P grounded power plug and 3 pins shrouded female IEC-320-C13 connector and is SJT 13A/300V rated.

The cord is available in many sizes from 3 feet to 15 feet. For the use of KIT, a 15 feet power cord will be used, making it easier for users to be able to operate K.I.T. further away from an AC outlet without having the need for an extension cord. Figure 27 shows a picture of this 16 Gauge AWG 3 prong power cable.

Figure 27 Three-Prong Gauge AWG Power Cable 16

3.4.3.2 Power Supply DC Output Connection

The type and size of output wires need to be taken under consideration for digital current wiring from the output of a power supply to the load. Ampacity and voltage drops are the two main considerations. American Wire Gauge, AWG, defines the diameter and cross-sectional area and the capacity of carrying of current of the wire. The voltage drop is the amount of voltage that is lost throughout the length of the wire due to the resistance of the conductor [31]. The sizing of the wires depends on the needed ampacity or voltage drop. In order to determine this, the power consumption at the load is first needed to be determined.

The power supply RSP-320-5 will be used to mainly power the 768 LEDs. The total power consumption of the LEDs is 230.4 Watts at 46.08 Amps. In this case, a minimum of an 8 Gauge AWG wire is required. To avoid voltage-drop, the 24 rows by 32 columns will be divided into eight of 3 rows by 32 columns LEDs. This means that each strip will have the total of 96 LEDs or 28.8 Watts or 5.76 Amps power consumption. Since the power supply will not feed the entire 46.08 amps to the LEDs, an 8 Gauge AWG wire will not be needed anymore.

According to the minimum recommended DC AWG cabling for protected outputs table in Table 14, a 16 Gauge AWG wire can be used instead as its current rating is up to 10 amps as highlighted. The use of larger diameter wires with a smaller AWG number will have the potential to reduce voltage drop and heat generated across the wires [31].

Table 14 Minimum Recommended DC AWG for Cabling for Protected Outputs

Total Power Module Current Rating (Amp)	**Wire & Lug Gauge (AWG) using 90°C wire**
5	18
10	16
15	16
20	14
30	12
40	10
50	8
75	6
100	2
125	2
150	(1) 1 AWG or (2) 6 AWG
175	(2) 4 AWG

The chosen wire as shown in Figure 28 is a high-purity oxygen-free copper core with a uniform silicone material for flexibility as shown in Figure 33 below. The wire is made of tinned copper with a 1.53 millimeters conductor diameter and 252 of 0.08 millimeters strands wire. The silicone rubber is soft and flexible, giving its overall diameter of 3 millimeters. It has a high temperature resistance of -60°Celsius to 200 °Celsius. The nominal voltage is 600 volts and can bear up to 12 amps. This 16 Gauge wire has a very low impedance, making its connections highly efficient. It is also very lightweight, which will be ideal for K.I.T.

Figure 28 Electrical Wire 16 Gauge AWG

3.5 Audio Controller

The following sections describe the different components considered for audio operation of the KIT. Ultimately audio is not part of the requirements for the design, however, it plays a major role in customer's gaming experience. Because of this there was extra effort put forth into considering different audio components while prioritizing other designs requirements. The following sections describe three different systems for audio control.

3.5.1 Adafruit Audio FX Sound Board

The Adafruit Audio FX soundboard comes with built 16MB storage on the board capable of storing up to 15 minutes of quality compressed audio. The board is equipped with a micro USB port to preload the music library. The provided code is compatible with compressed Ogg Vorbis or uncompressed WAV files. Capable of producing 44.1KHz 16-bit stereo high-quality sound. To control audio playback the board is equipped with 11 triggers. The backbone of the audio FX sound board is the VS1000 system-on-a-chip integrated circuit. Capable of decoding the Ogg Vorbis audio format this is superior to MP3 decoders, as it is free of royalty payments due to MP3 patenting. The chip also has NAND FLASH and Full Speed USB interfaces [32]. The schematic for the Adafruit Audio FX Sound Board is shown in figure 29 below.

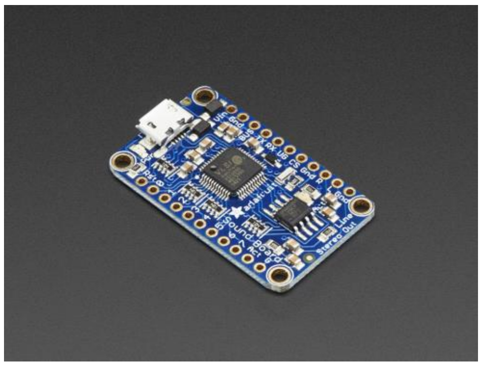


Figure 29 Adafruit Audio FX Sound Board

3.5.1.1 Adafruit Audio FX Sound Board VS1000 Ogg Vorbis Player IC

VS1000 is a single-chip Ogg Vorbis (license free audio codec) player and a system-on-a-chip (SoC) for various control and audio applications. This IC fits the usages of a simple K.I.T. gaming platform extremely well as it is capable of delivery high quality sound at extremely low costs. The main requirements for the audio are that it can be driven through UART communication, loop a single audio file, and switch audio files quickly. When the game logic deems it is necessary for an audio file to play the trigger information is sent through the UART pin to detail which audio file is to be played and at what volume.

The VS1000 decodes the audio file and transmits this signal to the audio jack built into the ADAFRUIT board. The VS1000 utilizes a DPS core for low-power and high performance. Features such as NAND FLASH interface, Full Speed USB port, general-purpose I/O pins, SPI, UART give this chip many communication capabilities. As for the audio capabilities this chip is equipped with a high-quality variable-sample-rate stereo DAC. The built in VS1000 firmware implements a default player that reads and plays files from NAND FLASH. NAND Flash is a type of nonvolatile storage technology that does not require power to retain data [33]. NAND is perfect for storing files replacing files frequently. With the USB interface the VS1000 firmware can use typical USB protocol or turn the USB into an Audio Device providing a single chip USB headphone application.

3.5.1.2 VS1000 Characteristics

Table 15 below displays the operating characteristics. This information is provided to understand the recommended settings as well as the limitations of this integrated circuit.

Table 15 VS1000 Operating Characteristics

VS1000 Operating Characteristics		
Parameter	Recommended	Max
Input Voltage	4 V	5.5 V
Analog Positive Supply	2.8 V	3.6 V
Digital Positive Supply	2.3 V	2.65 V
I/O positive supply	2.8 V	3.6 V
Input Clock Frequency	12 MHz	13 MHz
Operating Temperature	-	85 C
Storage Temperature	-	150 C
Input Current Limit	-	200 mA

3.5.1.3 Operation of ADAFRUIT Audio FX Board

The available tutorials for the ADAFRUIT audio FX board simplify the use of this component for the K.I.T. significantly. The audio files are loaded onto the storage of the board via USB interface. When plugged into any computer the board is recognized as a removable disk. There is no file structure needed for the disk, load the files onto the board exactly as you would with a flash drive. The board is powered directly off the K.I.T.'s 5-volt power supply. Alternatively, it is possible to power the audio component with three double AA batteries as the board comes equipped with a voltage regulator. The audio is controlled through the UART communication capabilities built into the VS1000. Arduino provides a simple library for communication commands such as pause, play, loop, and volume control. The audio files can be numbers to simplify desired audio file selection. Because of the built-in system, it is not necessary to configure audio decoding of any kind as the system will recognize the formats and decode them accordingly.

3.5.2 Teensy 3.2 and Audio Shield

Another audio option is the Teensy 3.2 and Audio Shield. Figure 30 below displays the module. This design comes with a micro-SD card reader allowing for a large library of different audio sources. The design has USB communication capabilities to receive audio cues from the raspberry pi. This combination allows for stereo headphone and stereo line-level output. The library paired with the Teensy 3.2 available from www.pjcr.com allows for input and output simultaneously together. It is possible to play multiple sound files, create synthesized waveforms, apply effects, mix multiple streams, and output high quality audio [106].

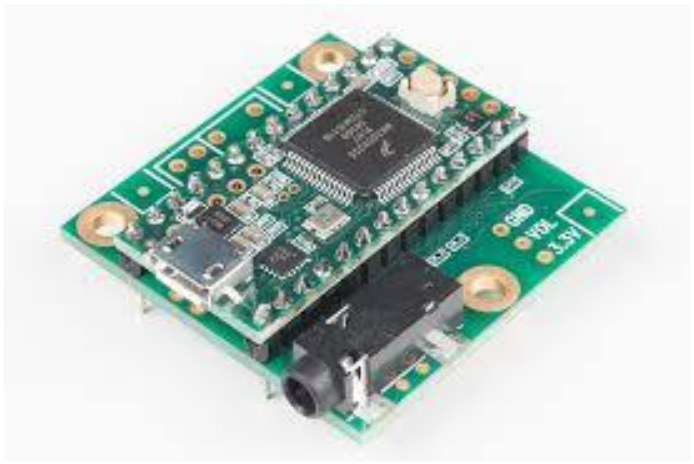


Figure 30 Teensy 3.2 and Audio Shield

3.5.2.1 Operation of Teensy 3.2 and Audio Shield

The audio system utilizes the I2C pins SDA and SCL to control audio output. The system utilizes three different clocks, the LRCLK, BCLK, and MCLK signals from the Teensy 3.2. The audio shield utilizes the SGTL5000 system in a chip to manage audio synthesizing. The SD socket is accessed with SPI communication to the Teensy 3. This board also allows for a potentiometer to be used to manipulate volume levels. The library provided by PJRC supports SD cards up to 32 GB in size. Finally, the Audio Shield comes equipped with a headphone jack to transmit audio signals.

3.5.3 Raspberry Pi 2

The last audio system design consideration is using the Raspberry Pi 2. The Raspberry Pi, shown in figure 31, comes equipped with audio functions. The Raspberry Pi is not designed for mixing audio signals; however, it has the baseline needed to create an audio mixing emulator. There is little documentation on using a Raspberry Pi as an audio mixer. The Raspberry Pi comes equipped with a headphone jack. The full capabilities of the Raspberry Pi are detailed in the game controller section as this component is chosen to handle gaming logic.

Figure 31 Raspberry Pi 2

3.5.4 Audio System Comparison chart

The proper audio components for the KIT was decided based on three main factors; Cost, Audio Mixing, and available documentation. The Teensy 3.2 and ADAFRUIT FX soundboard have extensive documentation toward creating an audio system. The ADAFRUIT FX is not capable of mixing multiple signals and has a smaller library available. The Teensy 3.2 has a larger possible library and is capable of mixing multiple signals. The raspberry pi is virtually free for this function as it is already chosen for game logic operation. Ultimately because of the available documentation and the ability to mix multiple signals, The Teensy 3.2 is chosen for audio. It should be noted that the audio is a side project that is not part of the main requirements for the KIT. This design is chosen with a lower priority to meet desired requirements and design is subject to change. The features of each components are shown in comparison in Table 16 below.

Table 16 Audio System Comparison

Parameter	**Teensy 3.2/Audio Shield**	**ADAFRUIT FX**	**Raspberry Pi 2**	**Unit**
Input Voltage	+3.5 ~ +5.3	+3.5 ~ +5.3	+6~+20	V
Power Consumption	0.3	0.3	1.2	Watt
Working Temperature	-25 ~ +80	-25 ~ +80	-40 ~ +85	C
Storage	32 GB	16 MB	64 GB	-
Signal Mixing	Yes	No	Yes	-
Headphone Jack	Yes	Yes	Yes	-
Cost	$33	$19.97	$29.99	US Dollars

3.6 Major Components

All of the major components needed to complete this project has been bought and are going through testing as discussed in the Testing section of this document. Figure 32 below shows the major components that have been ordered including all the microcontrollers, the power supply, buttons, LED strips, terminal rings, breadboard, required Gauge wires and more. There are still more components that have not been bought, such as framing materials, since they are not as important and do not required testing to be done.

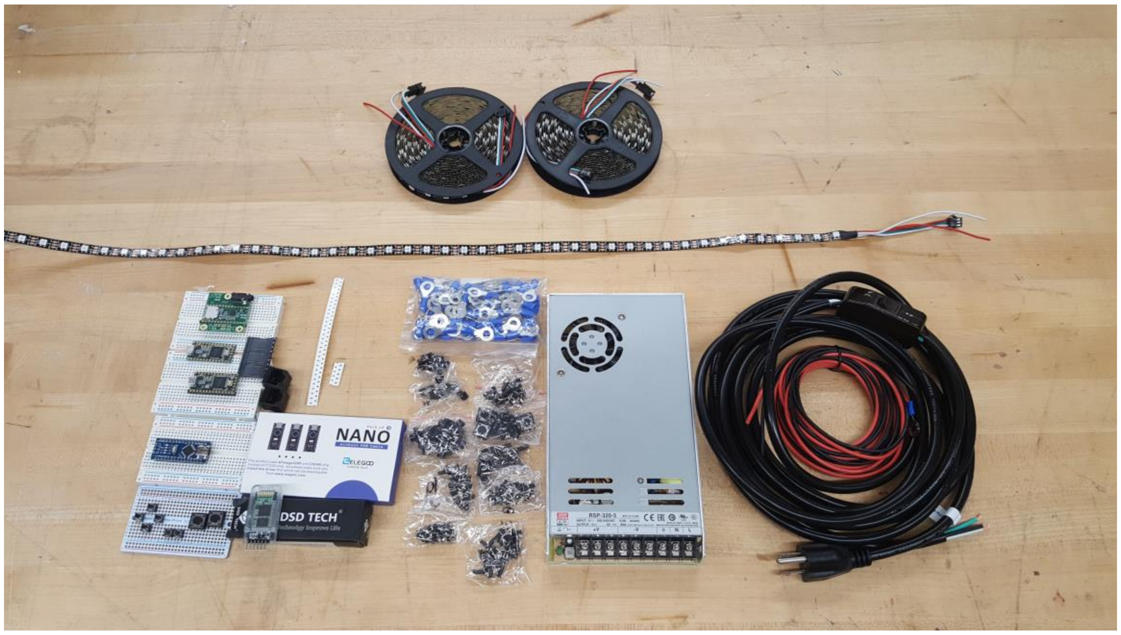


Figure 32 All Major Components

4. Design

The design includes both hardware and software designs. This section will describe all of the components needed to create the overall design of the project including the framing materials and the schematics of the KIT itself. Furthermore, some electrical wiring schematics are discussed within the section.

4.1 KIT Box

The final appearance of KIT should be sleek and simple in design with a black and red exterior. As per dimension, there are 32 LEDs wide by 24 LEDs tall, each LED in the strip is about 0.656 inches apart from each other, resulting in the LED screen being roughly $32 \times 0.656 = 20.992 \approx 21\ inches$ wide and $24 \times 0.656 = 15.744 \approx 16\ inches$ tall. The board will have a depth of at least 2.5 inches. There will be a divider between the LEDs, which will be in the front, and other components of KIT that will be on the back of the divider. An acrylic plexiglass will be placed above the LEDs to help diffuse the lights with foam board dividers placed between each LEDs to make the overall design more harmonious. KIT will also have a back stand to provide users with ease of usage without having to stand it against a wall or other platform. The overall appearance of the design is shown in Figure 33.

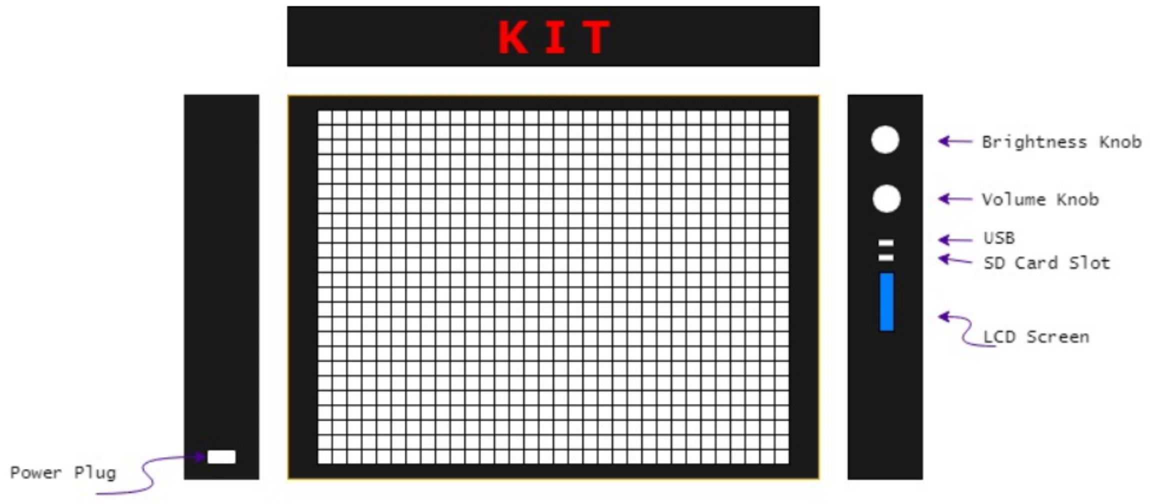


Figure 33 KIT Box Design

The power cord is connected on the bottom left side panel of KIT. The ON/OFF switch will be attached to the power cord between KIT and the AC outlet, making it easier for users to turn the device on and off without having to unplug any power cord.

Other functional buttons are placed on the right-side panel of KIT, including a brightness knob, volume knob, USB port, SD card slot, and an LCD screen. The brightness knob will be placed on top of the right panel. Users can adjust brightness of the LEDs by using this turntable knob. Right below it will be a volume knob. Followed by a USB port for controller connections. The SD card slot will be placed below the USB port. The gaming software is written onto an SD card and can be transferred to KIT’s memory via the SD card slot. Lastly, there will be a small LCD screen display at the bottom right panel of KIT that will display the internal temperature of the device, the version, debugging, and Bluetooth connection status. The LCD will also have a built-in user interface buttons for selection purposes.

4.1.1 Frame

Since the final design of KIT should be clean and simple, a few types of materials were being considered to be used as the frame of KIT including plywood, medium-density fiberboard (MDF), and foam core board. There are advantages and disadvantages to all of the materials, but plywood is selected as the material that will be used for the frame, the divider between the LEDs and other components of KIT, and also the back panel with a back stand. The discussions of each material are elaborated below.

4.1.1.1 Plywood

Plywood is a type of wood product, engineered by combining and pressing multiple layers of wood veneer together into one solid piece [34]. Depending on grades, plywood can come in smoother texture and prettier in appearance. The higher the grade, the more expensive the plywood will be. Although there will not be too much dust when cutting into, plywood may be a bit tougher to work with when it comes to cutting it into shape due to the multiple layers of wood veneers. Plywood is very strong and sturdy, great for holding screws in place since it carries a grain, making the bonding of pieces easy compared to the other materials. With its great ability to endure heat, it will not expand, contract or wrap even under extreme temperatures [35]. The thermal conductivity of plywood is shown to be 0.12 W/mK which is a decent value [36]. Plywood also works well in preventing the internal noise within the box and not escaping; this can help aid the human factor and ease of use for users.

Furthermore, plywood, shown in figure 34, is great for staining. One thing to keep in mind when working with plywood is the edges need to be polished and finished, as there are multiple layers of wood veneers pressed together. It is also best to make pilot holes prior to inserting any screws to ensure the accuracy of the screws.

Figure 34 Plywood

4.1.1.2 Medium Density Fiberboard

Medium density fiberboard is engineered and manufactured using very fine hardwood and softwood dusts. Similar to particle board, the fibers are glued with wax and resin and compressed under high temperature to produce an MDF. MDF is usually cheaper than plywood depending on the types and grade used. The board has a very smooth texture, free of knots, and can be easily cut into shapes, which is great for KIT simple design. MDF is heavier compared to plywood, it is denser but not as sturdy as plywood. It can be easily damaged if handled too roughly. According to a technical datasheet, an MDF board has a thermal conductivity of 0.15 W/mK [37]. While plywood is great for staining, MDF takes paints very well.

A big disadvantage of MDF, shown in figure 35, is the toxic resins, such as urea-formaldehyde and phenol-formaldehyde, used to produce the fiberboard itself [38]. Since the fibers are very fine, a high level of dust may occur and be inhaled when worked with. It is also hard to bond the pieces together with screws due to the fineness of the material causing it to split and chip easily.

Figure 35 Medium Density Fiber Wood

4.1.1.3 Foam Core Board

Compared to plywood and MDF, foam core board is much lighter in weight and easy to work with. It normally has three layers: polystyrene or polyurethane foam for the inner layer, and a white clay coated paper or cotton archival paper or common brown Kraft paper for the two outer layers on each side [39]. The board can be easily cut into shapes with a sharp knife. Its lightweight and a very low thermal conductivity, how easy heat can travel through a material, of roughly 0.025 W/mK [40], characteristics are ideal for KIT as we want the design to be portable and lightweight. However, it may be difficult to assemble the different panels together since some foam board do not adhere easily with glue. The use of screws for stronger bonds is also not possible to use with foam board.

Another concern with this type of material is that although foam boards, shown in figure 36, are made to be a strong material, they are not as durable as plywood and MDF. The foam board will most likely be easier to break with lower force compared to the other two materials, giving the design the cheaply made feeling. This issue combined with the inability to handle screws result in the foam board not being an ideal material for the project.

Figure 36 Foam Core Board

4.1.1.4 Framing Material Comparison

The chosen material for K.I.T.'s framing is plywood as the since it is sturdier than the other two materials and is also easier to work with. Table 17 can be referred to for easier comparison of the thermal conductivity of each material. The lower the thermal conductivity, the better it is for usage with heat. It can be clearly seen that foam core board has the best thermal conductivity, yet it is not chosen as the ideal material for K.I.T. for its lack of durability compared to plywood.

Table 17 Thermal Conductivity Comparison

Type	**Thermal Conductivity (W/mK)**
Plywood	0.12
Medium Density Fiber Board	0.15
Foam Core Board	0.025

4.1.2 LED Diffuser

The visual appearance of lights coming out of LEDs can make the overall design and gaming experience a bit distracting and uncomfortable. Because LEDs create invisible hot spots, bright spots of concentrated light, a diffuser can be used to eliminate this issue and help scatter the light to achieve a soft glowing light that better sooth the eyes. A diffuser sheet will also protect the fragile LED bulbs from outside substance, such as water, dust, and other small objects, that can damage the LEDs. The overall appearance will also look nicer and cleaner. There are a few types of materials being considered for the usage including an acrylic sheet, polycarbonate sheet, and polycarbonate film. The different types of materials are discussed further in the sections below including the advantages and disadvantages of each material.

Because we do not want the diffuser to completely block out the lights coming from LEDs, an acrylic plexiglass sheet in white would be the most suitable material for KIT. The thickness of the sheet is chosen to be 1/8 inch, just thick enough to be durable and impact resistant but not too thick that will block the lights. The chosen size is 24 by 24 inches since KIT display will be 21 by 16 inches, giving a bit of room for error.

4.1.2.1 Acrylic Plexiglass Sheet

Acrylic is considered one of a type of thermoplastic, a material that will be at its liquid state at their melting point of 130 degree Celsius [41]. Since the sheet will be placed about an inch above 768 LEDs, it should be able to handle the heat that will be produced while KIT is operating. It is made up of multiple molecules by combining a monomer with a catalyst together and molding into sheets with desired size and thickness [42]. Acrylic also has a decent impact resistant which is ideal for KIT, making it portable and users would not have to worry about breaking or ruining the overall appearance of the device. Plexiglass, shown in figure 37, is widely used for diffusing LED lights, for this reason along with others mentioned earlier, acrylic sheet is chosen to be used as the diffuser.

Figure 37 Plexiglass

4.1.2.2 Polycarbonate Sheet

Similar to acrylic, polycarbonate, shown in figure 38, is also a thermoplastic and has a high impact resistant, making it another good option to be utilized as a diffuser. It is made by combining catalysts and distilled hydrocarbon fuels together to form plastics. The mixture is then poured into a mold to get the desired shape, thickness, and size [43]. Furthermore, polycarbonate sheet is not as shiny as acrylic and is prone to scratches. Its melting point temperature is much higher than of acrylic’s, roughly 155 degree Celsius [43], although the LEDs should never come close to that high of a temperature. There are multiple grades of polycarbonate sheets that are available depending on the type of usage. A couple disadvantages to polycarbonate sheets, compared to acrylic plexiglass, are that they are harder to find and a bit more expensive.

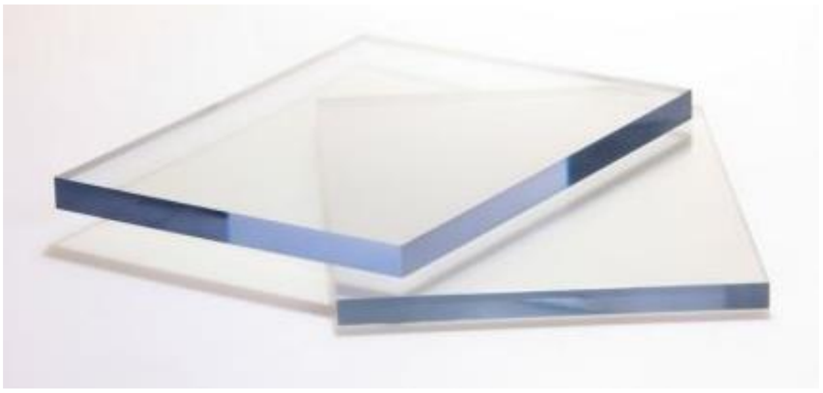


Figure 38 Polycarbonate Sheet

4.1.2.3 Polycarbonate Film

Polycarbonate film, shown in figure 39, is very much the same as polycarbonate sheet, except the final product is a thin film that is more flexible in contrast to a rigid polycarbonate sheet. It will also do a great job as an LED diffuser, similar to the other two material. The film also has a high impact resistant, making it another good option for KIT. However, because the film is made so thin, 0.005 inches to 0.03 inches [44], it is not a suitable LED diffuser as the final appearance will be quite flimsy and it will be a bit difficult to keep the film in place above the LEDs. Polycarbonate film is also hard to find and is not always available on some supplier websites.

Figure 39 Polycarbonate Film

4.1.3 LED Dividers

On top of using an acrylic plexiglass sheet to diffuse and scatter the LED lights, KIT will utilize dividers that will be placed between each LED to keep the light and color from each LED from blending in together. Since LEDs permit heat when lit up, a foam material is chosen as the preferred material to use for the dividers because of its low thermal conductivity of 0.034 W/mK [45], compared to a regular carboard that doesn't have as low of a thermal conductivity value of 0.5 [46]. Using this comparison, it is clear that a foam board will be better for insulating heat. On the other hand, a thin foamboard shares the similarity in firmness with cardboard. It is manufactured with a polystyrene foam core and paper sheets on the outside. The polystyrene foam core help keeping the foam in shape as it bounces back after being handled, making it resilient to crushing.

The LEDs will be placed about 0.656 inches apart, this means the thickness of the divider should not exceed 3/16 inch, to provide enough space for the LED pixels. Since KIT's dimension of the LED screen is approximately 21 by 16 inches, the chosen size of the foam board is 18 by 24 inches, leaving plenty of room for errors. The divider itself will be approximately 1 inch wide by 21 inches long to cover the entire area of KIT's screen. Slot spaces will be cut on the dividers so that they can be connected together and surround the LEDs. After comparing a few brands of foam board, ELMERS is chosen as the manufacturer to go with. A picture below shows exactly what the dividers will look like including their dimensions.

4.1.4 Separation Sheet for LEDs and Other Components

The LEDs will be resting on a platform made out of a conductive material, to distribute the heat from the LEDs away from the front panel into the back panel where there will be fans blowing the heat out of the device. An aluminum sheet will be used as it is lightweight and has the conductive ability.

4.1.4.1 Aluminum sheet

Aluminum, shown in figure 40, is a type of metal that has the ability to conduct electric current. It is a very lightweight metal that has a specific weight of 2.7 gram per centimeter cubed [47]. This type of metal naturally generates a coating of a protective oxide that is highly corrosion resistant, making it shelf-life last longer than other metals. Aluminum is also a good reflector for visible lights [47], which will help reflect the lights from LEDs and make it appear brighter through the diffusion glass.

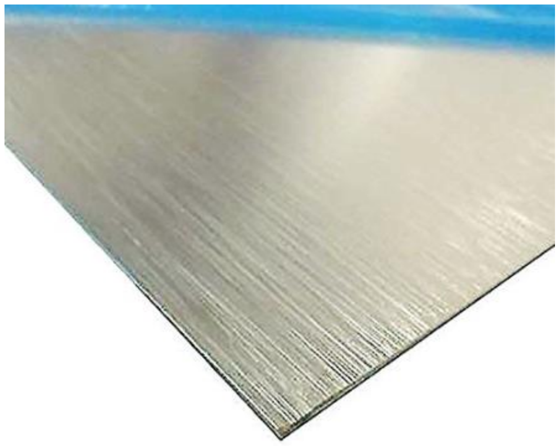


Figure 40 Aluminum Sheet

Furthermore, aluminum is a very good heat conductor, almost twice as good as copper. The thermal conductivity of pure aluminum is about 235 watts per kelvin per meter [48].

When alloys are added to pure aluminum, the conductivities will slightly decrease. The higher the thermal conductivity, the faster the heat can be transmitted. This is why it will be used to separate the LED strips and other components to transfer the heat from the front panel, where the LEDs will be, to the back panel of K.I.T. where there will be fans installed to blow the heat out of the device and keep the overall temperature relatively cool. Care should be taken since the aluminum sheet will be very thin and can be bent easily.

4.1.5 Framing Connection

The frame of K.I.T. will be made of plywood which will need to be fastened together using an adhesive. Some of the options include the use of wood screws, dowels, and wood glue. There are both advantages and disadvantages between all types of adhesive that are discussed in the following sections. After some research, it has been decided that K.I.T. will utilize dowels with the combination of wood glue to use as connections between plywood pieces.

4.1.5.1 Screws

Wood screws are commonly used to connect pieces of woods together and strengthen the joints. The thread comes in several types including thread forming, sharp crested, and coarse pitch. The sizing of the threads is varied from 32 threads per inch with the diameters being 0.06 inches, or #0, to 7 threads per inch with the diameter or 0.372 inches, or #24. Wood screws come in various length as well, from a quarter of an inch to up to 6 inches [49]. The materials that wood screws are commonly made of include steel, stainless steel, brass, and silicon bronze.

A set of screws may be used to fasten the plywood pieces together to form the framing of K.I.T., however, since plywood is made of pressing multiple layer of woods together, the wood itself may split if wood screws are not applied correctly. A pilot hole should be predrilled to prevent the wood from splitting when the screws are installed. Some wood glue may also be needed to put on the joints to have a stronger bond between the screw and the plywood. For the reason that the wood has a high chance of splitting, wood screws may not be the best option for fastening the plywood pieces together.

4.1.5.2 Dowels

Another method of connecting pieces of wood together is by using dowel joints. Dowels are round pins made out of wood with small diameter. They come in various size and shape as shown in Figure 41. They are, sometimes, used to fasten the joints instead of nails or screws. Dowel joints are the strongest type of joints when done correctly, and they help to create a neat and clean finish [50]. To connect the wood pieces together using dowels, precise markings should be made prior to drilling the dowel pins. A dowel plate is a steel plate that has holes of different size depending on the dowel size used, it can be used to making the drilling of dowel pins easier. One of the disadvantages of dowels is that it is easy to make mistake such as misalignment. After the dowel pins are successfully made and connections are made to the right alignment, wood glue should be applied to the joint to strengthen the connections. With the use of dowels, the final appearance of KIT should be very clean on the outside.

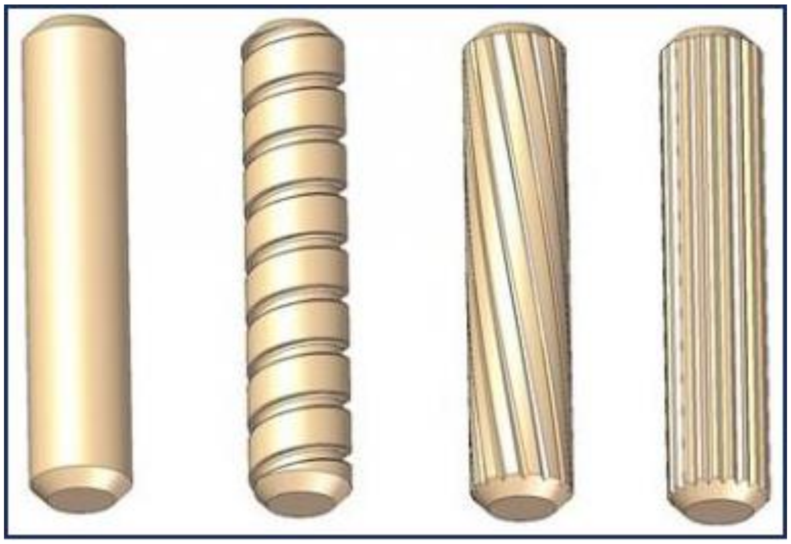


Figure 41 Wooden Dowels

4.1.5.3 Wood Glue

Wood glue will be used to strengthen the joints of K.I.T.'s frame. The type of wood glue used should be considered depending on the desire of the finished look and function. There are multiple types of wood glue available on the market including, but not limited to, Polyvinyl acetate (PVA) glue, hide glue, epoxy, cyanoacrylate (CA) glue, and polyurethane glue [51]. The type of wood glue selected should be heat resistance and should not interfere with the finished look when dried. PVA glue is most common when it comes to woodworking. It provides a very strong bond especially when the wood pieces are clamped while the wood is drying. Using the combination of wood glue and dowels should provide KIT with strong and stable connections throughout the entire device.

4.2 KIT Hardware Design

This section details the necessary hardware components for the overall KIT design including the powering methods, audio, and display designs. The schematic below shows the system architecture of the KIT Design. What is not shown in this schematic is the power and ground contacts for each subsystem. Each system is powered by the same 5-volt power supply. The power and ground for each system is detailed greater in their respective sections. Figure 42 displays the system architecture design schematic of the overall system.

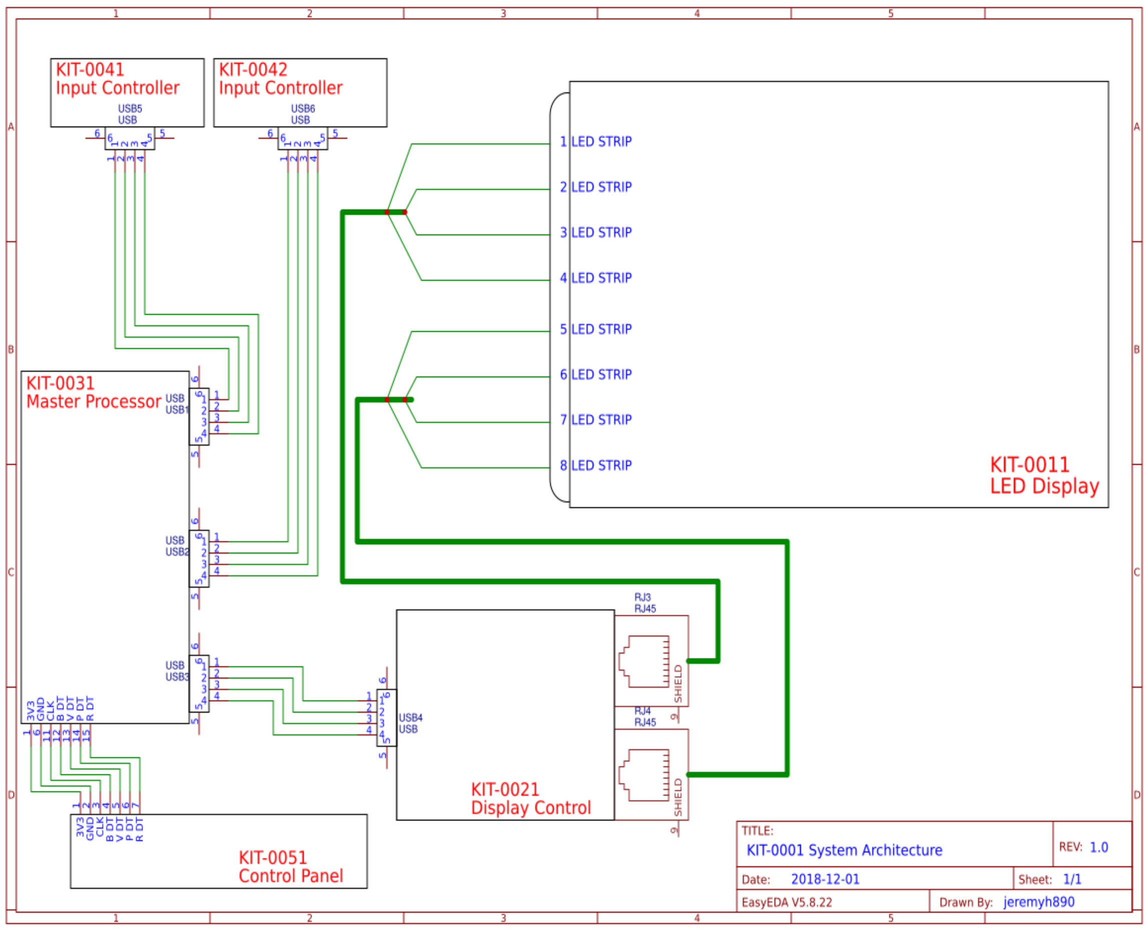


Figure 42 KIT Architecture Schematic

The control panel is made up of the LCD display, volume control, brightness control, and power button. The main communication between the different control units is the USB interface. The master processor is a raspberry pi 2 computation system, the display control is driven using a Teensy 3.2 and the Input controllers are driven using Arduino development boards. Each subsystem is described in greater detail in the following sections.

4.2.1 KIT ON/OFF Switch

Users will have the ability to turn K.I.T. on and off via a switch instead of having to unplug the power cord from an AC outlet. Multiple methods of connecting a power switch were thought of and discussed in the sections below. The decision was made to have the switch connected between an AC power outlet and the RSP-320-5 power supply. The other option that was not chosen is having the switch connected between the power supply and the device. This is thought to be not as safe as the other option because the power supply will constantly be connected to the AC outlet even though K.I.T. is turned OFF. By doing this, the power supply never really stops working since it will constantly convert the AC power to DC power even though there is nowhere for the DC power to go. This could lead to the power supply being overloaded or burnt out for being on for too long.

By having the switch connected between an AC power source and the RSP-320-5 power supply, when users switch the switch to OFF, there will not be any power going from the AC outlet to the power supply. This prevents the power supply from burning out for being on for too long.

4.2.1.1 Switch Between Outlet and Power Supply

The first option of an ON/OFF switch is by connecting an inline switch between the AC power outlet and the RSP-320-5 power supply. The implementation of this switch, shown in figure 43, will make the powering of K.I.T. easier from users' perspective, simply toggle the switch instead of having to completely plug or unplug the power cord. The switch that is chosen for the project is from the company Leviton, model 5410. It is a heavy-duty cord switch, single pole feed-through switch rocker, and is rated for 3-amp 125 AC volts, perfect for usage with the selected power supply. The dimension of this switch is 1.3 by 2.8 by 5 inches.

Figure 43 Leviton Single-Pole Inline Switch

The inside of the switch is shown in Figure 44. Since the three-prong power cable is selected for connecting the power supply to an AC outlet, the outer jacket of the cable will need to be first removed. Then, the black live wire will be cut and stripped. According to the instruction sheet, the leads of the black wire will then be connected to the terminal screws of the switch by twisting the strands of each lead together very tightly, then loop them clockwise about three-fourth turn around the terminal screws [52]. The screws should then be tightened firmly over the wire loops to 12-14 in. lbs., and any excess wire strands that are not secured under the screw should be cutoff and removed. This is to be done for both terminal screws.

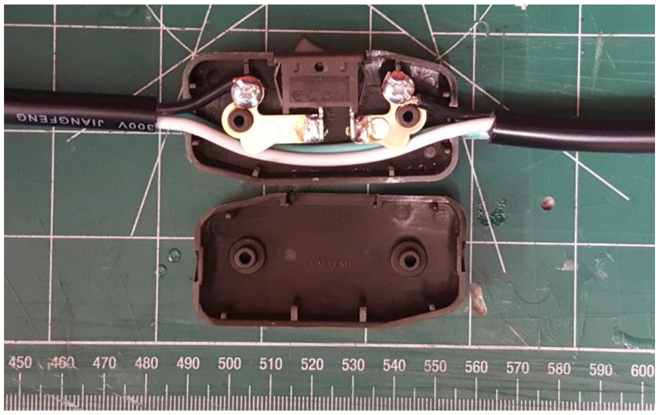


Figure 44 Inside of Leviton Single-Pole Switch After Assembled

4.2.1.2 Switch Between Power Supply and Device

The other option of an ON/OFF switch application for K.I.T. is by connecting the switch between the RSP-320-5 power supply and the LEDs strip terminals. In this particular case, the switch will be a turning knob switch instead of regular toggle switch, although it is very much the same principle. The knob would be made of wood, to match K.I.T.'s framing design. Two metal pieces, any electrically conductive metal, will be attached to the wooden block using an adhesive. The end of the metal pieces is bent, leaving small spaces between the two pieces. The turning knob will have a small conductive screw attached to the bottom, when the knob is turned clockwise, or to ON position, the screw will close the gap between the two metal pieces, enabling the electricity flow from the power supply to K.I.T., thus turning the device on. To turn the device off, the wooden knob would be turned counterclockwise, opening up the gap between the two metal pieces and stop the electricity flow between the power supply and the device. The wiring diagram for this switch can be seen in Figure 45 below.

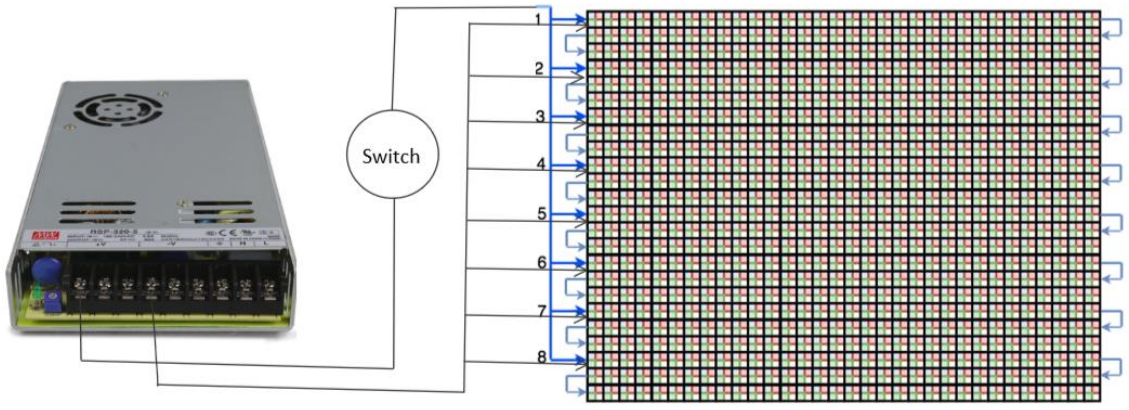


Figure 45 ON/OFF Switch Between Power Supply and LEDs

This knob would be attached to the side panel of K.I.T. and spray painted black like the rest of the frame. Although this is a great homemade switching project and is relatively cheap, it requires a lot of wood work and is not as safe as the other option since the current flow through this switch would be as high as 47 amps if all the LEDs are turned on at their brightest. Also, as discussed in the previous section, the power supply would constantly be fed AC power and converting it to DC power, making it work constantly if the power cord is not unplugged from an AC outlet. This could lead to the possibility of burning out the power supply, which is not an ideal situation. For this reason, K.I.T. will have an ON/OFF switch that is between the AC outlet and the power supply.

4.2.2 Adafruit Pi Plate LCD Design

The LCD design is intended to aid in testing and verification of K.I.T. operation. If errors occur in game, the LCD screen will be used to display the last statement echoed from the code. This allows designers to pinpoint the location of problematic code. Additionally, the LCD screen will display the internal temperature of the KIT gaming platform. This serves the purpose of proving to the customer that operating temperature requirements are met. If the temperature gets to hot a warning symbol will flash on the LCD display. If the temperature begins to enter a failure range the LED display will shut down, the game will pause, and a “Critical Temperature” message will scroll across the LCD display.

4.2.3 LED Brightness Adjustable Knob

Users will have the ability to adjust the brightness of the LED display according to their liking using a potentiometer knob on the side of K.I.T.’s panel. One of the options for dimming LEDs is by connecting an LED dimmer between the power supply and the LED strips. When the knob is turned counterclockwise to OFF, the LEDs will actually be completely off. The more clockwise the knob is turned, the brighter the LEDs will appear. This is due to the adjusted voltage from the power supply being allowed to be delivered to the LEDs, the higher the voltage, the brighter the LEDs will be.

After more research is done and the decision has been made to go with WS2812B LED strip, the dimmer switch is no longer applicable for K.I.T. as the WS2812B has a built-in constant current control no matter what the input voltage is. This results in a software encoded dimming capability. This will be accomplished by adding a simple rotary encoder. The encoder will receive user input via turn knob. The information is relayed to the OCTO2811 LED library which holds a variable for brightness adjustments.

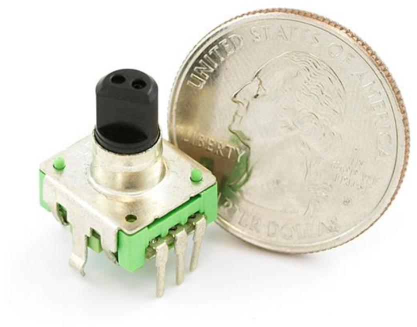


Figure 46 Rotary Encoder

The ALPHA 381-ENC130175F-12PS rotary encoder chosen for the K.I.T., shown in Figure 46, relays information via gray code. As can be seen, it is smaller than the size of a quarter. In the code there is a predefined encoder state table. When the encoder is turned in a clockwise direction the input received is referred to as a +1 state. When the encoder is turned in a counterclockwise direction the input received is referred to as a –1 state. The software uses this information to change a variable value that holds the brightness of the LED display from 30%-100%. Through testing it is was determined that any lower than 30% brightness makes the display unusable for game play. There are two receiving ports reading the state of the encoder. When the encoder is turned counter clockwise port B cycles from low to high. When the encoder is turned clockwise port B cycles from high to low. This state of port B is compared to the state of Port A to decide if the encoder was turned clockwise or counter clock wise as Port A will indicate the same cycle regardless of direction. The encoder is operated using interrupts alongside the game logic. To reduce conflict in operations the interrupt pauses gameplay prior to receiving input and gives the user a reasonable amount of time to make brightness adjustments. The gameplay is resumed once the encoder state fails to change for more than 3 seconds.

4.2.4 Audio Design

Audio is the unsung hero of all video games. Adding sound to the K.I.T. is imperative to ensuring the player becomes immersed into the gameplay. Whether it is the excite the player in faster pace gameplay, reward the player for an amazing accomplishment, or force the player to hear the sounds of defeat, audio drives the emotions the game intends to deliver. However, adding audio to the platform complicates both the gaming logic and the hardware design. Instead of focusing simply on how the game is displayed on the K.I.T. now there must be synchronized sound. To simplify the design, instead of having sound effects for user inputs there will be an allotment of sound files per game. This means that the audio effects that are tied to each game are referred to as background music. To simplify it further, when a user is in normal gameplay audio file 1 will be playing, dependent on the game a second or third sound can play but not simultaneously. If file 1 is played first and file 2 is set to play, file 1 will end, and file 2 will begin. This reduces costs to the consumer as audio mixers significantly complicate the code design.

The K.I.T. utilizes an Adafruit Audio FX Sound Board to add sound effects to gaming. With 16 MB of storage this soundboard can hold up to 15 minutes of 16-BIT audio. The board comes equipped with 11 triggers, which will allow the playback of 11 different audio files. The audio board supports both OGG and WAV audio file formats. To further simplify the design the output will be via a TRS (headphone) port. With this port the user can connect headphones or a personal speaker with built in amplifying capabilities.

4.2.5 Display Design

The KIT gaming platform utilizes a multitude of components to allow for fast refresh rate, dimmable capabilities, reliable operation, and ultra-bright output. The overview of the design is shown in figure 47 below.

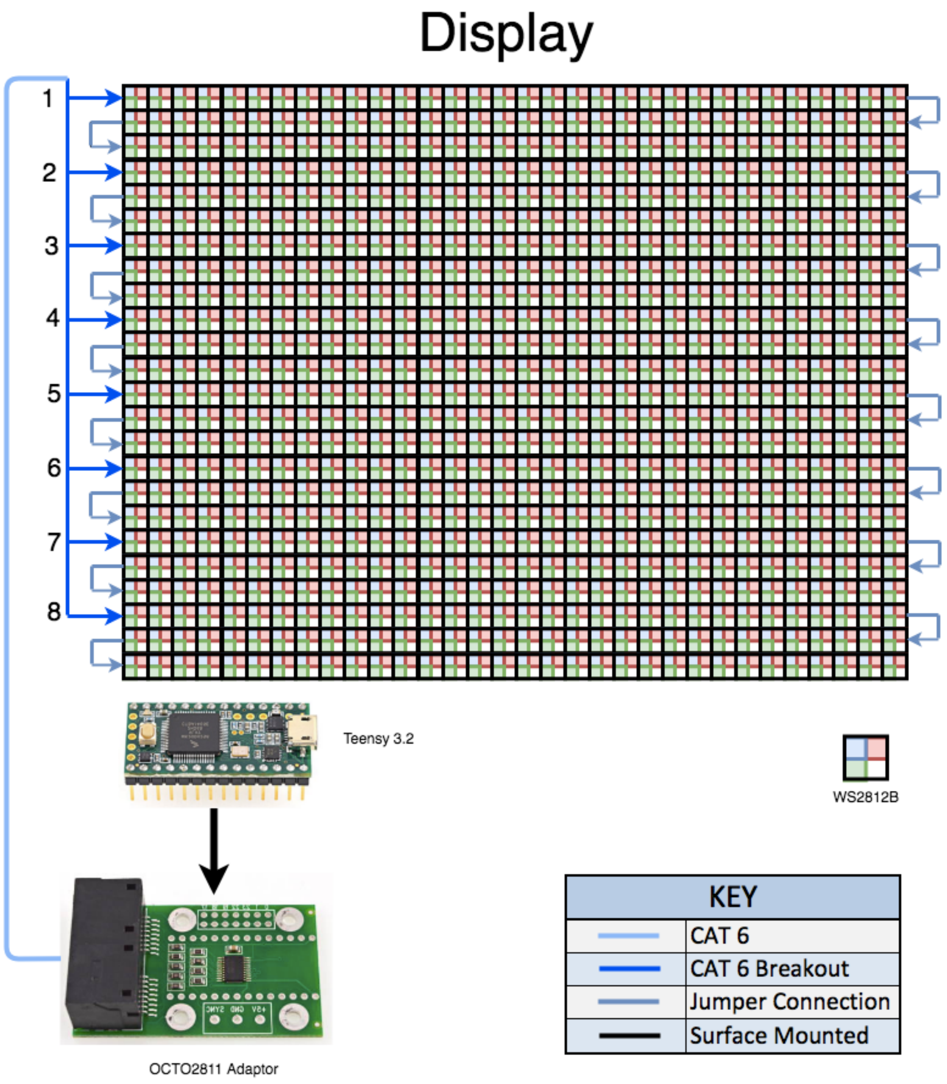


Figure 47 Display Diagram

As seen, there is a matrix of WS2812B LED components. These LEDs are individually controlled RGB light sources. Each red, green, and blue light source is driven by an integrated circuit, which is receiving commands through a unique serial peripheral interface. This interface will be discussed in more detail in the WS2812B section. The integrated circuits within the WS2812B receive their commands from a Teensy 3.2 microcontroller. The Teensy 3.2 microcontroller achieves an exceptionally fast refresh rate with the rated CPU speed of 72 MHz, however, to deliver the proper signal voltage levels the OCTO2811 adaptor is used. These two components are the main drivers of the LEDs. The KIT utilizes a Raspberry Pi microcontroller to relay game display information to the Teensy 3.2 via USB protocol. This Raspberry Pi microcontroller is not considered a main component of the display as it handles the game logic and user inputs for the KIT. The WS2812B and Teensy 3.2 capabilities, specifications, and advantages are more detailed in their respective research sections.

4.2.6 Light Emitted Diodes WS2812B

World-Semi is the original manufacturer of the WS2812B LED strip. The display accomplishes a resolution of 24x32. Comprised of a control circuit and RGB chip integrated into a package of 5050 components to form a complete control of pixel point. Each pixel comprising of the three primary colors achieve 256 brightness full color display. The K.I.T. requires that each LED is individually addressable which the WS2812B provides. This is accomplished by an internal intelligent digital port data latch and signal reshaping amplification drive. This integrated circuit takes the place of the LED power drivers that communicate the proper lighting to each LED such as the Texas Instrument's TLC5940. Not only does the WS2812B's control circuit come integrated into each LED, the strip allows addressing of multiple LEDS through the signal reshaping amplification drive.

4.2.6.1 Timing

This design utilizes eight strips of 96 LEDs connected in series. In the display diagram shown in figure 47, in hardware design section, it can be seen that there are eight different signals coming into the display matrix. At connection 1 the display signal enters the data-in port of the first LED, which is cascaded to the data-in port of the 96th LED via the two jumper connections. This is repeated for each of the 8 connection points. The WS2812B datasheet indicates a 300-nanosecond maximum delay after each LED. At the 96th LED there will have been 95 such delays, which yields a total strip delay of 28.8 microseconds. Every 24-bit message takes 30 microseconds.

With 96 LEDs, each strand will take 2958.8 microseconds to update. This along with the 50-microsecond reset delay to update the display yields a refresh rate of 333 Hz for each 96 LED strip. This is not how fast the system is capable of updating the display; instead it is the limit of the LED strip capabilities. The actual limit of the display is governed by the microcontroller. However, this information indicates that with the proper set-up a refresh rate of 60Hz is easily possible. Further research provided independently by engineering enthusiast Tim of "cpldcpu.wordpress.com" indicates the LEDs recognize a reset at anything greater than 9 micro-seconds which would allow for a faster refresh rate. Table 18 compares the results of Tim's experiments with the information provided in the datasheet.

Table 18 Timing

Timing WS2812B		
Time Periods	Tim's Results (ns)	Datasheet(ns)
HI_IN "0"	62.5-563	400
HI_IN "1"	> 625	800
Period In	> 1063	~1250
Delay out	208	300
Reset	> 9000	> 50000

Table 18 describes how a bit of 0 or a bit of 1 is communicated to the WS2812B internal circuit. The period of each bit is approximately 1250 nano-seconds. In this period if the signal is high for less than approximately 500 seconds the bit is 0 as show by the HI_IN "0" row. If the signal is high for greater than 625 nano-seconds or approximately 800 nano-seconds the bit is 1. This type of transmission is referred to as NZR transmission. Figure 48 below is a diagram of the data transmission protocol

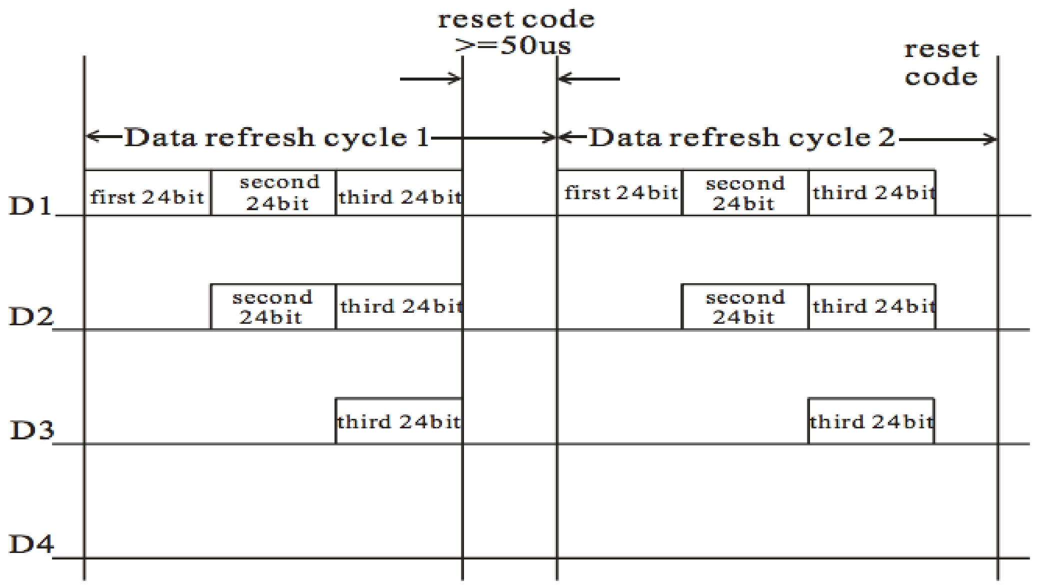


Figure 48 Transmission Protocol

4.2.6.2 LED Protocol

Each LED latches onto the first 24 bits it receives and then passes every bit following until a reset delay is recognized. Then the cycle repeats. Figure 48 from the datasheet shown above details this communication transmission with cascaded LEDs referenced as D1, D2, D3. Each LED's output is defined by a 24-bit message. The composition of the message is shown below in Figure 49. The transfer protocol is MSB first. With G7-G0 for the green LED, R7-R0 for the red LED, and B7-B0 for the blue LED.

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0

Figure 49 BIT Composition

4.2.6.3 Operation Ratings and Characteristics

The table 19 below describe the different operating characteristics of the WS2812B LED strip. Power rating characteristics are also shown in the table below. It can be seen that the power supply voltage is rated between 3.5 and 5.3 volts.

Table 19 Absolute Maximum Ratings

Prameter	Symbol	Ratings	Unit
Power supply voltage	V_{DD}	+3.5~+5.3	V
Input voltage	V_I	-0.5～VDD+0.5	V
Operation junction temperature	Topt	-25～+80	℃
Storage temperature range	Tstg	-40~+105	℃

The environmental electrical characteristics are shown in table 20 below. This information is used to understand the maximum and minimum current and voltage levels the LEDs can handle.

Table 20 Electrical Characteristics
(TA=-20～+70℃, VDD=4.5～5.5V,VSS=0V,unless otherwise specified)

Prameter	Smybol	conditions	Min	Tpy	Max	Unit
Input current	I_I	$V_I=V_{DD}/V_{SS}$	——	——	±1	μA
Input voltage level	V_{IH}	D_{IN}, SET	$0.7V_{DD}$	——	——	V
	V_{IL}	D_{IN}, SET	——	——	$0.3\ V_{DD}$	V
Hysteresis voltage	V_H	D_{IN}, SET	——	0.35	——	V

The switching characteristics are shown in table 21 below. This information is used to understand the delays between each LED component.

Table 21 Switching Characteristics

Prameter	Symbol	Condition	Min	Tpy	Max	Unit
Transmission delay time	t_{PLZ}	CL=15pF,DIN→ DOUT,RL=10KΩ	——	——	300	ns
Fall time	t_{THZ}	CL=300pF,OUTR/OU TG/OUTB	——	——	120	μs
Input capcity	C_I	——	——	——	15	pF

Finally, the RGB light emittance characteristics are shown in table 22 below. This information is used to understand the light emittance capabilities of the LEDs.

Table 22 RGB IC characteristics parameter

Emitting color	Model	Wavelength(nm)	Luminous intensity(mcd)	Voltage(V)
Red	13CBAUP	620-625	390-420	2.0-2.2
Green	13CGAUP	522-525	660-720	3.0-3.4
Blue	10R1MUX	465-467	180-200	3.0-3.4

4.2.6.4 Features and Benefits

The LEDs used for the WS2812B offer low driving voltage, environmental protection and energy saving. The luminescence characteristics are described as high brightness, large scattering angle, and good consistency. The control chip integrated within the LED allows for a simpler circuit design, smaller overall volume, and convenient installation. The IC includes an intelligent digital port data latch and signal reshaping amplification drive circuit, and a precision internal oscillator. The IC operates using a 12V voltage programmable constant current control; effectively ensuring the pixel's point light color height is consistent. Within the circuit there is an intelligent reverse connection protection, a shared power source for the LED and the control circuit. Capable of a complete 16777216 color full color display, and scan frequency not less than 400Hz/s. The input and output port transmission allow for a cascading design.

Finally, the design allows for data transmissions speeds of 800Kbps. The WS2812B selected for this project is compared to similar commercially available products such as the SK6812 in the table below. The main deciding factors for moving forward with the WS2812B was the extensive research information, design examples, and open source code libraries available for the WS2812B [53].

4.2.7 Teensy 3.2 Development Board

The Teensy 3.2 development board was chosen specifically to utilize the OctoWS2811 LED library. OctoWS2811 is a high performance WS2811 & WS2812 & WS2812B LED library, written by Paul Stoffregen, featuring simultaneous update to 8 LED strips using efficient DMA-based data transfer. The K.I.T. requires a 30 Hz refresh rate for smooth motion display. When operating 768 LEDs it is difficult to ensure that each strip is updated simultaneously at a high refresh rate. That is where the Teensy 3.2 Development board truly excels compared to competitive options such as the Arduino-Mega.

With a 72 MHz processing capability, the Teensy 3.2 is capable of receiving new display information, setting up the proper display LED matrix, and transmitting the information to the LED display with refresh rate capabilities greater than 60Hz [57]. This is also possible because of the 256kB of flash memory. The Teensy 3.2 microcontroller comes with 64kB of RAM, however 16 kB is likely enough to handle nearly all LED display libraries, the extra 48kB allows the Teensy to store addition code for potentially expanding capabilities. Figure 50 details the physical appearance and pin information of the Teensy 3.2 [57]. The specifications are broken down into greater detail in the following sections.

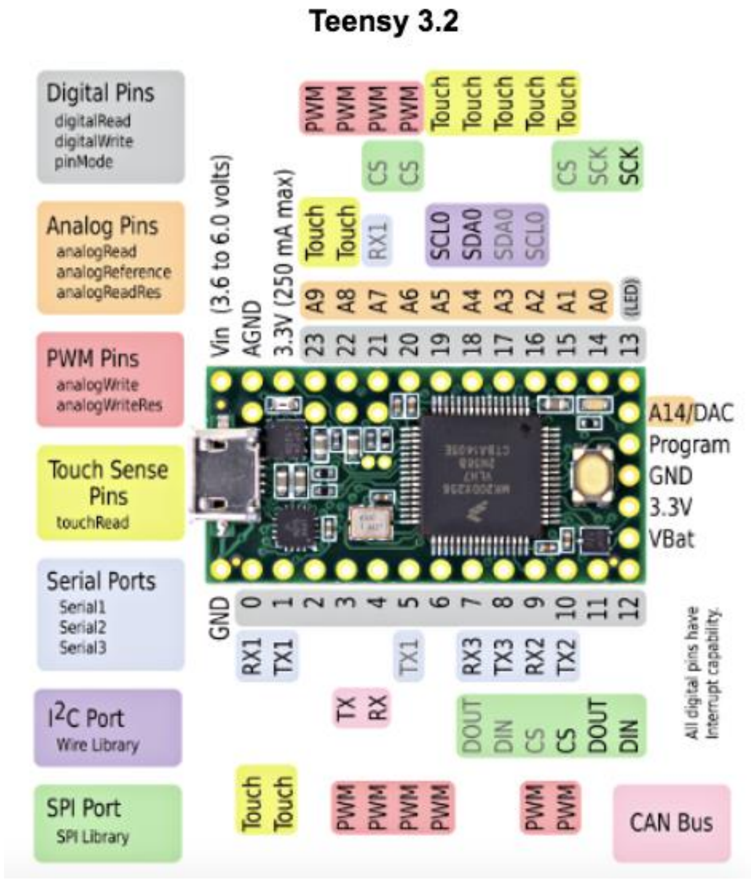


Figure 50 Teensy Diagram

4.2.7.1 Teensy 3.2 Operation Details

The Teensy 3.2 paired with the OCTOWS2811 library allows for ultrafast refresh rates. OctoWS2811 is designed for highly efficient data output to WS2812B-based LEDs, able scale to very large LED arrays. As mentioned in the LED section, each LED uses 24 bits, with each bit cycle requiring 1.25 µs, for a total of 30 µs per LED in the strip. By separating the LED strips into 8 equal lengths the time required to update the display is reduced provided the hardware can handle an update of 8 outputs nearly simultaneously. Driving 8 LED strips simultaneously allows each strip to be only 1/8th the length. All LEDs update 8X faster than driving only a single long strip. This design can potentially update 1000 LEDs in 3.8 ms, which allows a theoretical update rate of 240 Hz. This is accomplished by utilizing DMA channels with PWM outputs [54].

OctoWS2811 uses Direct Memory Access (DMA) to create the WS2812B waveforms with nearly zero CPU usage. Because the CPU is free, and interrupts remain enabled, the processor is free to receive data or perform computations in preparation for the next frame of display, while the previous one is in transit to the LEDs. The 8X faster update and free CPU time defines the key differences between OctoWS2811 and other libraries. OctoWS2811 uses 3 DMA channels to synthesize the WS2812B waveform. The hardware events, which trigger the DMA channels, are a pair of PWM waveforms [54].

The rising edge (both PWM rise at the same moment) triggers channel #1, which copies a fixed byte (0xFF) to an I/O register which sets all 8 output bits, causing the WS2812B waveform to begin each bit. The first falling edge triggers DMA channel #2, which copies one byte of the actual frame buffer data to all 8 pins. The bits, which are low transition to low at the correct time to create a zero, bit to each WS2812B LED. The bits that are high have no effect, because channel #1 already set all 8 pins high. The 3rd DMA channel triggers at the second falling PWM edge, causing all the WS2812B bits to be written to zeros [54].

The pins which were left high by channel #2, become low, as required by the WS2812B timing for a one bit. The pins which were already low are not changed. Together, these 3 I/O updates create a WS2812B waveform automatically without any CPU activity. Figure 51 below illustrates how the WS2812B waveform is synthesized. The end result of this harmony of synthesizing is that all 8 WS2812B waveforms are created with perfect bit cycles while significantly reducing CPU usage [54].

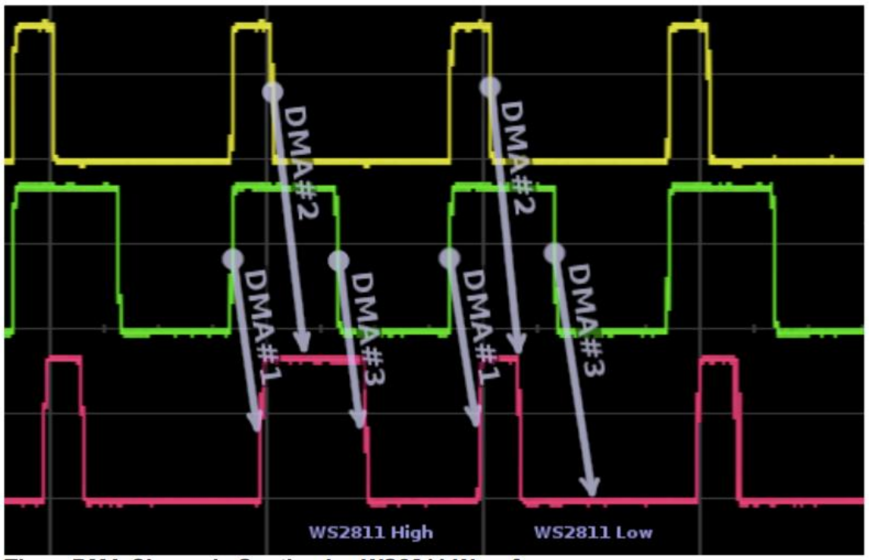


Figure 51 Utilizing DMA for WS2812B Synthesizing

4.2.7.2 Sophisticated Data Allocation

The Teensy 3.2 utilizes a crossbar switch allowing 4 separate bus masters to access 4 different bus devices simultaneously. Only when 2 bus masters wish to access the same device at the same time is one master made to wait. OctoWS2811 utilizes the dual-bus RAM and crossbar switch for maximum efficiency. The display memory is declared with "DMAMEM", which forces that memory to be allocated in the lower half of the RAM. USB buffers are also allocated in the lower RAM. The CPU accesses both, but the heavily used stack memory for local variables and temporary data is located in the upper RAM, which minimizes wait-inducing conflicts between the CPU and DMA. During normal operation, the CPU primarily accesses the upper RAM through a dedicated channel in the crossbar switch, which the DMA is able to read the lower half RAM and write to the I/O registers, also through crossbar switch paths that do not impact the ARM CPU's access to code and the most commonly accessed data. Of course, the USB also utilizes its own built-in DMA to transfer incoming data very efficiently into the lower RAM area. Figure 52 illustrates how the crossbar switch dataflow is organized [55].

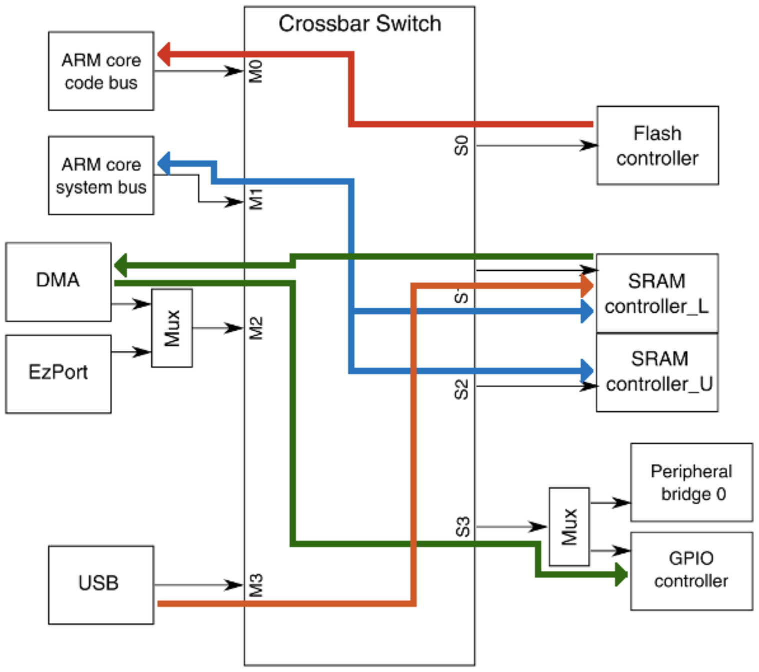


Figure 52 OctoWS2811 use of Dataflow on Teensy 3.2

Extremely efficient synthesis of the tight-timing WS2812B waveforms, with almost zero impact to the CPU is the primary benefit of OctoWS2811. The fast 32-bit CPU is fully available to prepare the next display frame. Interrupts remain enabled during transfers, allowing serial protocols like DMX to work [56].

4.2.7.3 MK20DX256VLH7 Processing Specifications

The Teensy 3.2 CPU is the MK20DX256VLH7 shown in figure 5 below. This processor has an operating frequency of 72 MHz. The static random-access memory is capable of storing 64 kB of information. The built-in flash memory is capable of storing 256 kB of information. The processor has 40 general-purpose input output (GPIO) pins, 3 universal asynchronous receiver-transmitter (UART) pins, and 12 Pulse Width Modulation (PWM) pins [57]. The full break down of the processor’s capabilities are shown in the development board comparison table below. The importance of this processor is the PWM pins, which are used to communicate to the LED display and the UART pins, which are used to receive display information from the raspberry pi development board.

4.2.7.4 Features and Benefits

The Teensy 3.2 provides an exceptionally fast clock rate for the cost. The Teesny’s 256 kB built in flash memory provides a perfect microcontroller for the LED display. The

K.I.T. display requires precise timing, which is accomplished by the Teensy's 72 MHz clock rate. The Octo2811 library requires the Direct Memory Access engine provided by the Teensy. The Arduino Mega could communicate with the LEDs without the need for a buffer to amplify the signal, however, the full cost of the Teensy and Buffer is lower than the Arduino. This along with the desire to use RJ-45 receptacles was a major driving factor for choosing the Teensy.

Also, there are a total of 3 UARTs (serial ports). Common communication capabilities such as SPI, I2C, I2S, CAN Bus, and IR modulator [55]. The I2S can be used for high quality audio interface. If a long-term clock is necessary, the configuration for a Real Time Clock is available with a user-added 32.768 crystal and battery. In addition to the USB there are 16 DMA channels. All of which come in the small Teensy package with dimensions of 1.4 x 0.7" (~35 x 18 mm). Table 23 below compares the Teensy 3.2 to the Arduino Mega in capabilities applicable to the K.I.T. design.

Table 23 Development Board Comparison

Development Board Comparison		
Parameter	Arduino Mega	Teensy 3.2
Operating Voltage	5V	3.3
Input Voltage	7 - 12V	3.6 - 5 V
Clock Speed	16 MHz	72 MHz
Digital I/O Pins	54	40
PWM Pins	15	12
UART Channels	4	3
Analog Input Pins	16	21
DC Current per I/O Pin	40 mA	25 mA
DC Current for 3.3V Pin	50 mA	-
Flash Memory	128 KB	256 KB
SRAM	8 KB	64 KB
EEPROM	4 KB	2 KB
Clock Timers	6	7
Supports I2C	X	X
Supports I2S	-	X
Supports CAN	-	X
Supports SPI	X	X
Cost	$38.50	$20

4.2.8 74HCT245 Buffer Specifications

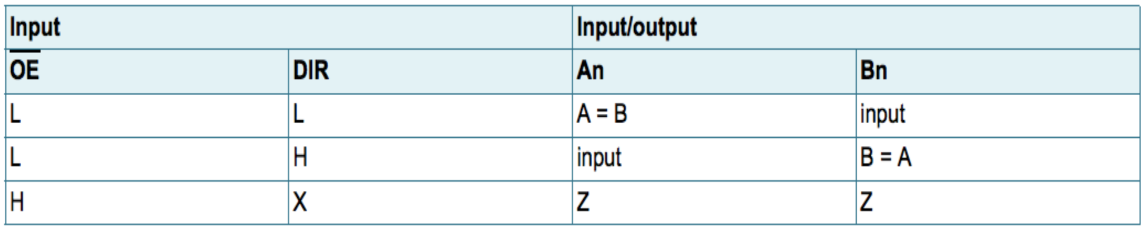
The 74HCT245 is used to boost the 3-volt output signals from the Teensy 3.2 to the necessary 5 volts required by the WS2812B integrated circuit. The custom display controller PCB features a 74HCT254 buffer chip and 100-ohm impedance matching resistor. This buffer chip amplifies the outputs of the Teensy 3.2 without signal degradation. The display controller comes is equipped with two RJ-45 Network ports allowing the use of Ethernet cable. Using a CAT6 cable in this port provides excellent benefits to signal integrity. CAT6 cables are designed for exceptionally high-quality high bandwidth signals while allowing minimal cross-talk between twisted pairs [55]. Between each connection of the 74HCT254 buffer chip to the RJ-45 Ethernet port is a100 ohm resistor. These resistors reduce signal ringing. Additionally, this RJ-45 port allows the LED display to be easily connected and disconnected for maintenance.

The 74HCT245 device is an octal non-inverting buffer/line driver/line receiver designed to be used with 3−state memory address drivers, clock drivers, and other bus−oriented systems. The device has non-inverting outputs and two active−low output enables. The purpose of this buffer in this design is to take the 3.3-volt output of the Teensy pins and amplify it to a 5-volt output for the LEDs. This is accomplished using 136 FETs built into the integrated circuitry of the buffer chip [58]. Conveniently the 75HCT245 has 16 input/output pins perfect for this application. Figure 11 below displays the pin and logic diagram of the 74HCT245.

4.2.8.1 74HCT245 Operation Details

The main purpose of the 74HCT245 driver/buffer chip is to boost the output signals of the Teensy 3.2. An added benefit is that the buffer isolates the impedance of the LED strips from the Teensy 3.2 ensuring effective signal integrity. The 74HCT245 accomplishes this signal amplification by utilizing a 5-volt VCC. The function table shown below indicates that with the proper setup of the enable pins 1 and 19 the output voltages will be approximately the same as VCC. The simple component configuration reduces delays to ensure the LED display can still accomplish high refresh rates. When 5 volts is applied to direction pin 1, the outputs are the B_n pins and the inputs are the A_n pins. The output voltage is boosted to the VCC voltage of 5 volts. The logic gates ensure that the output is 5 volts when the input is high and 0 volts when the input is low. Table 24 below is the function table.

Table 24 74HCT245 Function Table



Input		Input/output	
$\overline{OE}$	DIR	An	Bn
L	L	A = B	input
L	H	input	B = A
H	X	Z	Z

4.2.9 Game Controller Layout

For the game controller design, ten buttons are needed to perform certain commands. The overall layout of the game controller is shown in figure 53 below.

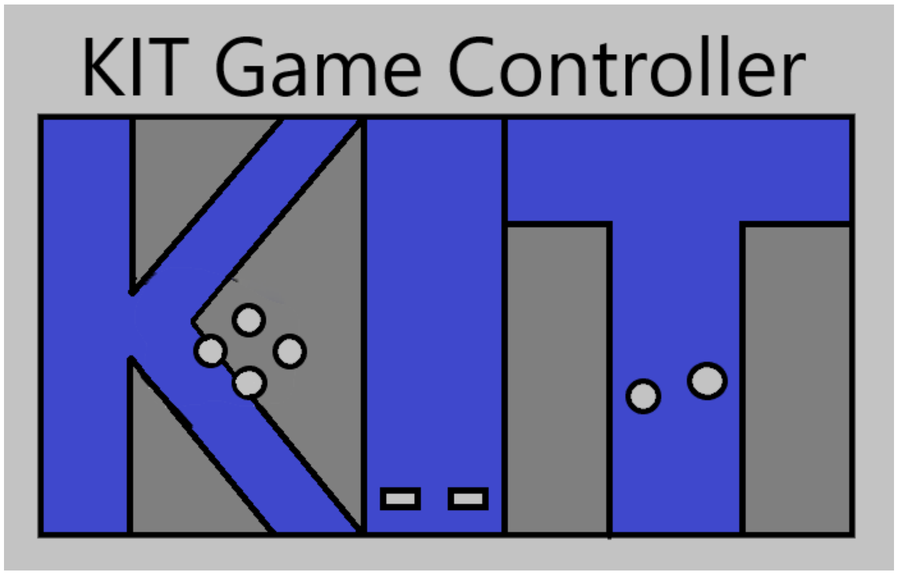


Figure 53 KIT Game Controller Design

The four buttons on the left will be used for basic directional control. One of the middle buttons will be used to pull up a menu while the other will be used in various ways depending on the game being played. One of the buttons on the right will be used for selection and the other button's function will differ depending on the game being played. As more games can be programmed to run on the LED game board, the controller's buttons can also be programmed to perform any function needed for the game.

4.2.10 Game Controller 3D Models

This section details the 3D models that were designed for the outer shell of the game controller. Each piece of the shell is detailed and displayed below. The figure 54 and 55 shows the top piece of the controller shell. This part matches the controller layout from an earlier section.

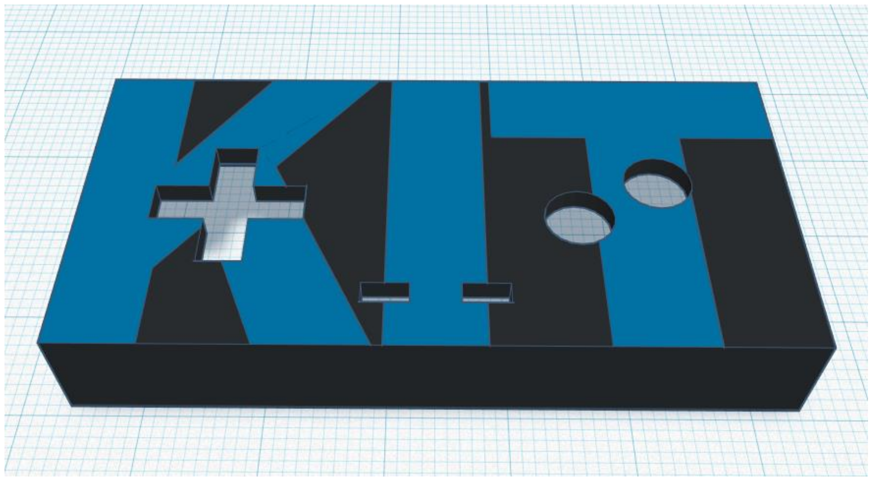


Figure 54 Front of Game Controller Shell Top

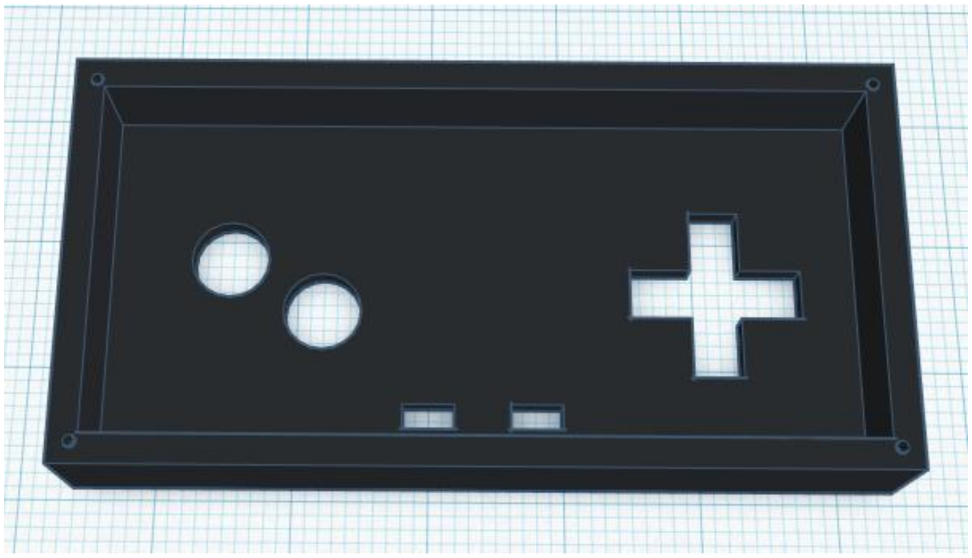


Figure 55 Inside of Game Controller Shell Top

The game controller shell top has inserts for the D-pad, two inserts for the square buttons and two inserts for the round buttons. There are four holes in the corners to screw the game controller shell bottom onto the top. The figures 56 and 57 the 3D model of the square buttons for the game controller shell.

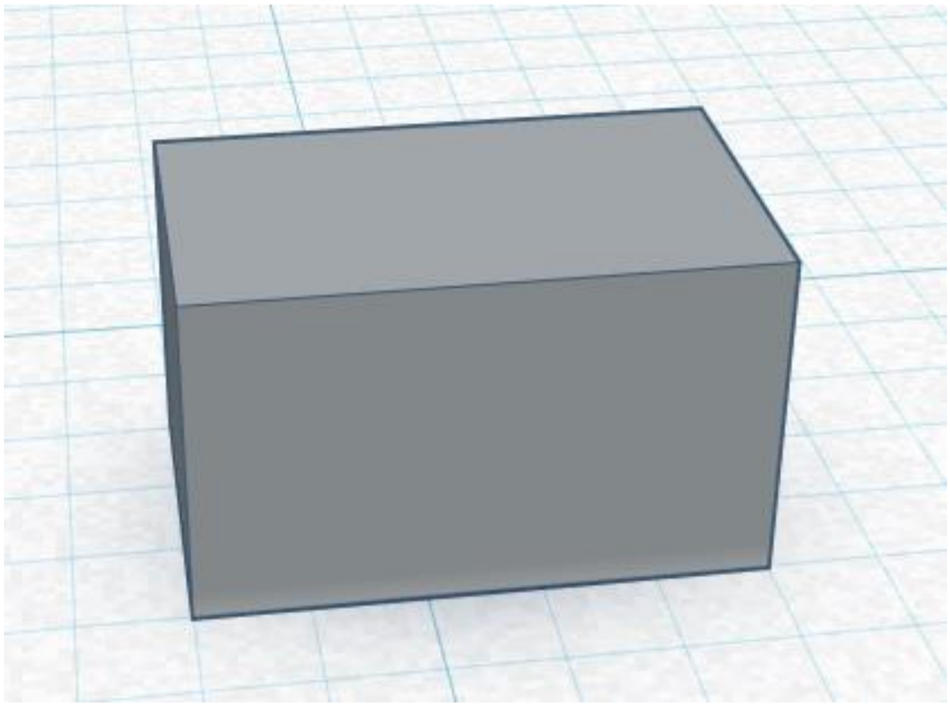


Figure 56 Top of Square Button

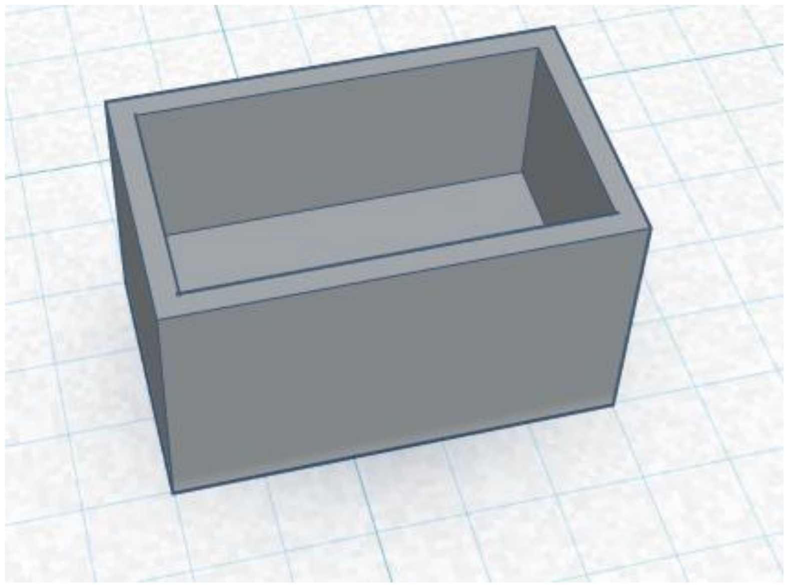


Figure 57 Inside of Square Button

Two square buttons will be printed and used to cover the two square tactile buttons on the breadboard. The material type for the button was chosen as TPE filament as discussed in an earlier section.

The figures 58 and 59 show the 3D model of the D-Pad for the game controller shell.

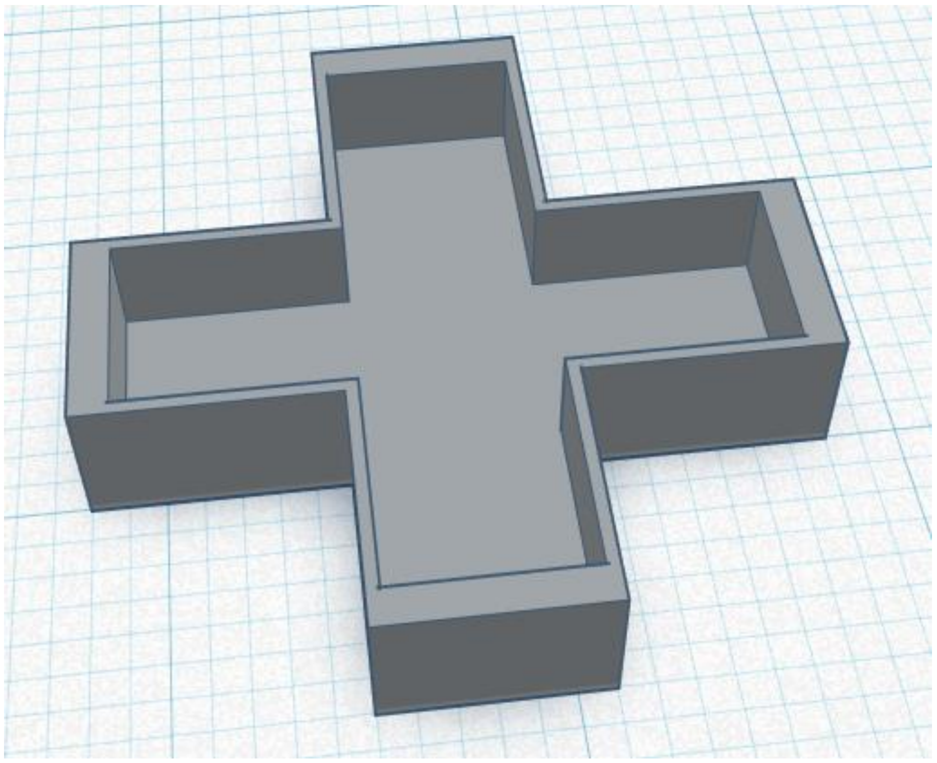


Figure 58 Inside of D-Pad

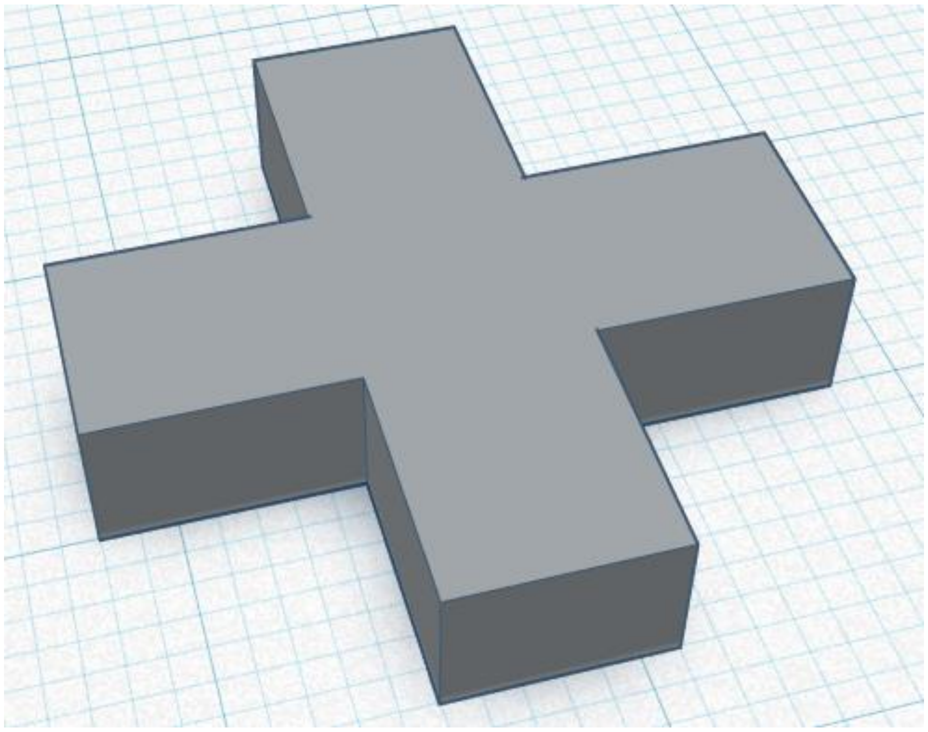


Figure 59 Top of D-Pad

One D-Pad cover will be printed using the discussed filament type. There will be four tactile buttons housed underneath this cover. Alternatively, a circular D-Pad was designed to allow a quick means of changing the game controller design if needed. The figure 60 below shows this alternative D-Pad.

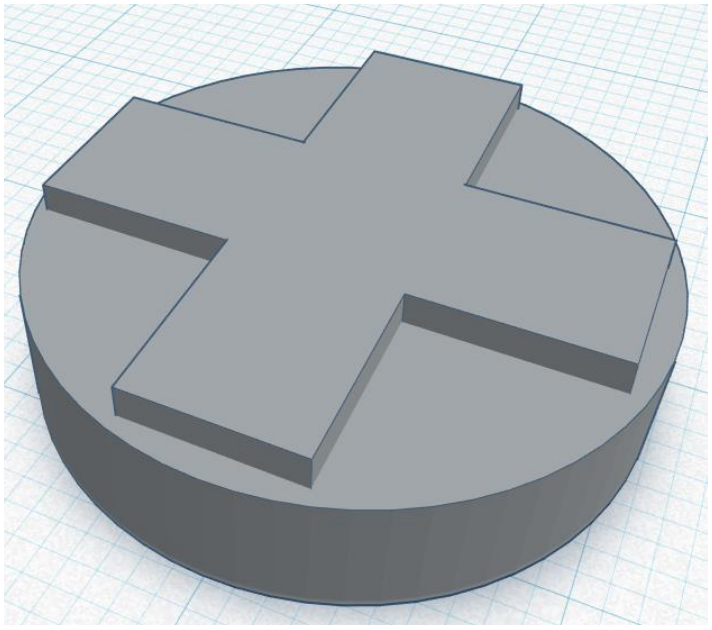


Figure 60 Alternative Circular D-Pad

Figure 61 below shows the 3D model of the round button for the game controller shell.

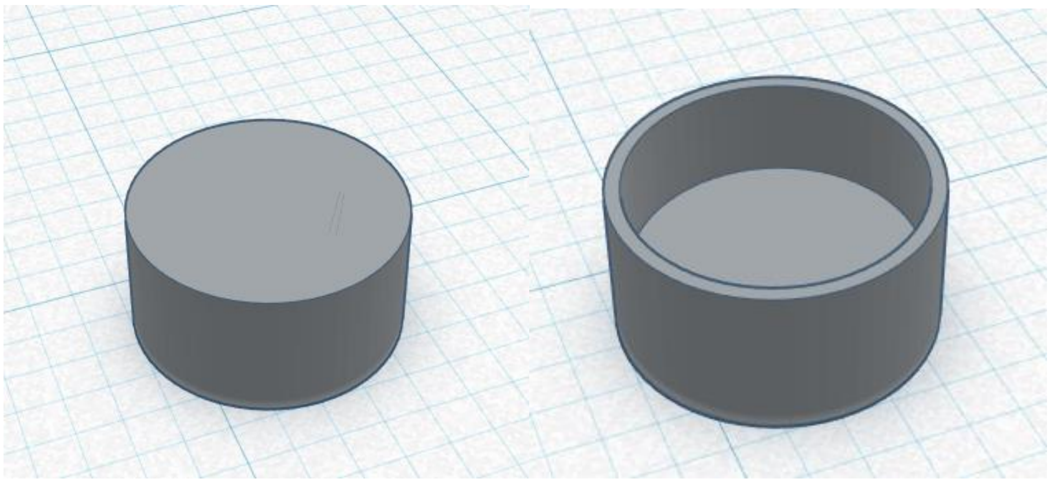


Figure 61 Top and Inside of Round Button

Two round buttons will be printed and used to cover the two round tactile buttons on the breadboard. The material type for the button was chosen as TPE filament as discussed in an earlier section.

Figure 62 below shows the game controller shell bottom. This piece will be screwed to the top and will help house the controller's components.

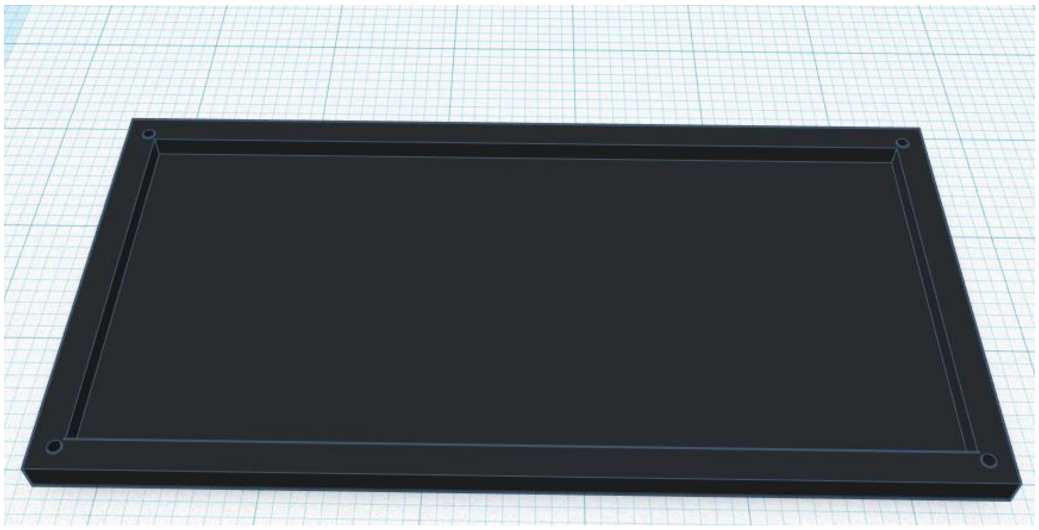


Figure 62 Back of Game Controller Shell

The inside portion of the game controller shell bottom will be attached to the Arduino Nano and HC-05 and the breadboard that is connected to the tactile buttons.

The fully assembled model of the game controller shell is shown in figure 63 below.

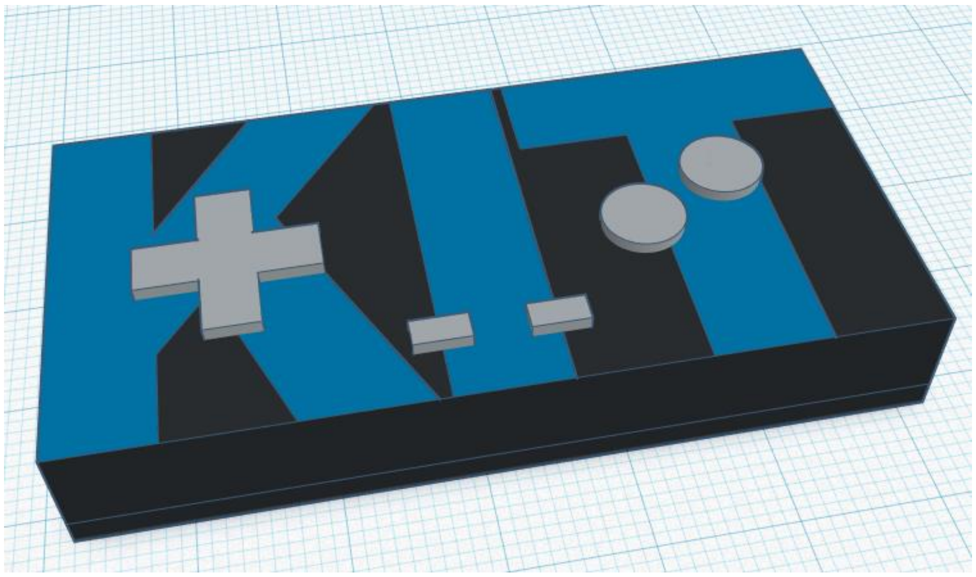


Figure 63 Assembled Game Controller Shell

The pieces of the game controller shell were modeled individually and saved as custom parts. The custom-made parts for the buttons and D-Pad were used to create the holes in the top of the controller shell. The parts were then fitted into place and assembled to create the original game controller design. The dimensions of each part are displayed in table 25 below.

Table 25 Controller Part Dimensions and Printing Material

Part	Dimensions (mm x mm x mm)	Filament Type
Round Button	6 x 6 x 12	TPE
Square Button	3 x 6 x 2.5	TPE
D-Pad	18 x 18 x 12	TPE
Top	120 x 90 x 15	PLA
Bottom	120 x 90 x 15	PLA

4.2.11 Cooling Fans

Since heat is one of the concerns for KIT because of the high number of LEDs inside the box itself, several cooling fans will be needed to keep the temperature under control. Another thing to consider when choosing the type of fan is the noise level because we also want to keep the noise as quiet as possible as to increase the human factor for users' comfort.

One of the candidates is the ARCTIC F12 Silent Case Fan, shown in figure 64, at a low price of $8 per piece. This model has a low fan speed which leads to it producing very little noise; the new alloy and lubricant combination also plays a factor in the reductions of friction and noise. The rated noise level is as low as 0.08 Sone compared to the 0.3 Sone of the other models. The fan speed is rated between 360 to 800 RPM while the airflow is at 37 CFM/62.9 meter cubed/hour [59]. At 5 volts DC, the fan speed is 270

RPM, this is really low and may not actually be useful for temperature control. The fan is capable of either blowing warm air out of the case or drawing in cool air into the case. Its dimension is about 120 millimeters by 120 millimeters by 25 millimeters and only weights about 3.84 ounces.

Figure 64 ARCTIC F12 Silent Case Fan

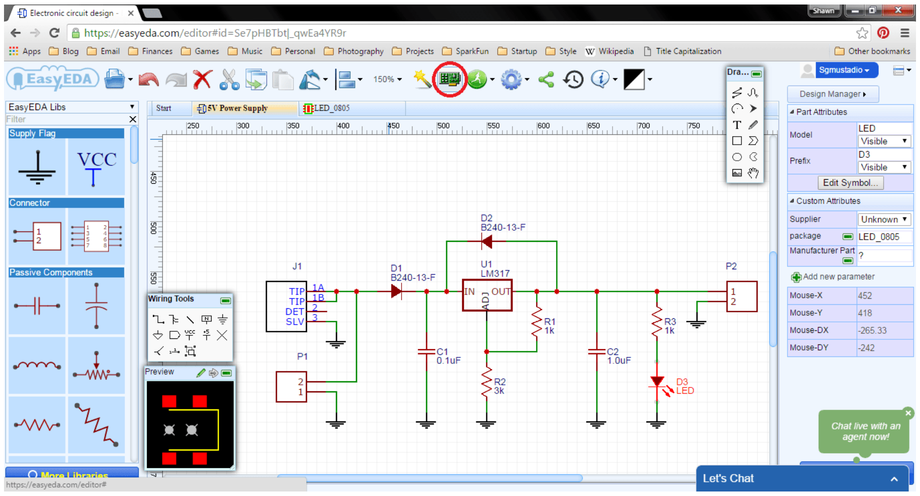
Another option for the cooling fan is the GDT8010S12B from the company GDSTIME. Its dimension is much smaller compared to the one of ARCTIC F12 Silent, 80 millimeters by 80 millimeters by 10 millimeters. Its rated voltage is 5 volts DC at 0.2 amps and the startup voltage is 3.2 volts. The rated current varies depending on the speed RPM. At the lowest speed of 1600 RPM, the current is as low as 0.18 amp. At the speed of 2200 RPM, the current is rated as 0.26 amp, and at the highest fan speed of 2800 RPM, the current is rated to be 0.48 amp [60]. The noise level is rated at 25 dBA. GDT8010S12B contains 11-blade fan that are supposed to increase the cooling efficiency at lower speeds. This case fan shown in Figure 65 is a little more expensive at $11 per piece, but it is much bigger than the ARCTIC F12. It is also lighter in weight at 1.23 ounces.

Figure 65 GDT8010S12B Cooling Fan

After more research, the ARCTIC F12 Silent needs a DC power supply input of 12 volts which we do not have. Though it would be a great option for KIT's use, the GDSTIME shown in figure 64 above, will be a better suited for KIT.

4.2.12 Printed Circuit Board Designs

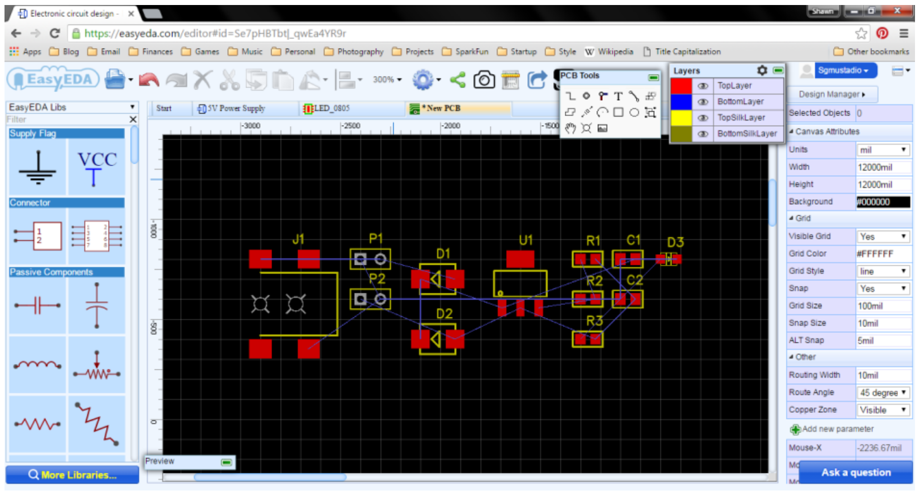
The Printed Circuit Board for the K.I.T. is designed using EasyEDA. EasyEDA is an electronic design automation software. The process of design begins by outlining a wiring schematic of the hardware components for the K.I.T. Figure 66 below is an example of this type of schematic.



Model: LED, Visible
Prefix: D3, Visible
Supplier: Unknown
package: LED_0805
Manufacturer Part: ?
Mouse-X: 452
Mouse-Y: 418
Mouse-DX: -265.33
Mouse-DY: -242

Figure 66 Wire Diagram Schematic

Once all hardware components and connections are known the design of the PCB can begin. From the wire diagram schematic, the EasyEDA software can import the schematic components into the PCB design schematic. Figure 67 below displays a PCB design.



Selected Objects: 0
Units: mil
Width: 12000mil
Height: 12000mil
Background: #000000
Visible Grid: Yes
Grid Color: #FFFFFF
Grid Style: line
Snap: Yes
Grid Size: 100mil
Snap Size: 10mil
ALT Snap: 5mil
Routing Width: 10mil
Route Angle: 45 degree
Copper Zone: Visible
Mouse-X: -2236.67mil

Figure 67 PCB Design Schematic

Connecting each component is easy with the blue guide line that comes from the wire schematic. Connect traces between the different points utilizing the different layers. Connecting the traces through the different layers of the PCB to is made easy with the layer hotkeys EasyEDA implements. After the PCB design is finalized the dimensions must be verified so that they fit properly within the physical case of the K.I.T. This is verified by a team review cycle that determines design is ready for manufacture. After the PCB is manufactured the components are surface mounted using soldering techniques.

4.2.12.1 Handheld Game Controllers PCB

The Printed Circuit Board for the game controller was also designed using EasyEDA. EasyEDA is a free online PCB design software that you run in your web browser. It has libraries of components that were added in by EasyEDA and has component contributions from users.

The schematic for the game controller was designed first. In order to design what was initially discussed in the game controller layout section, eight pins on the Arduino Nano would have to be connected to push buttons. The HC-05 Bluetooth module would then be connected to the Nano using resistors with a 5V power supply needed to power the entire circuit. This schematic design is displayed in the figure 68 below.

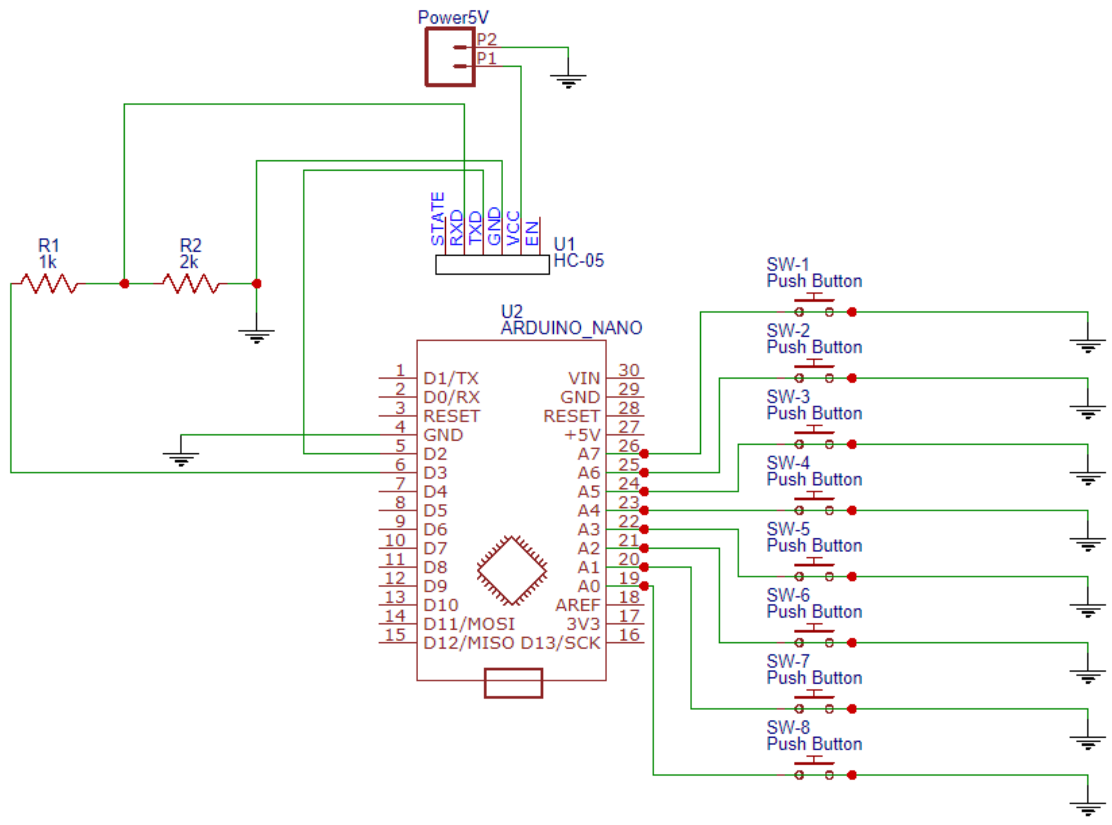


Figure 68 Designed Game Controller Schematic

After designing the schematic, it was exported to a PCB schematic where the connections shown in the schematic had to be made. The components were laid out on the PCB board in a way that matches the original game controller layout design. The completed game controller PCB design is displayed in the figure 69 below.

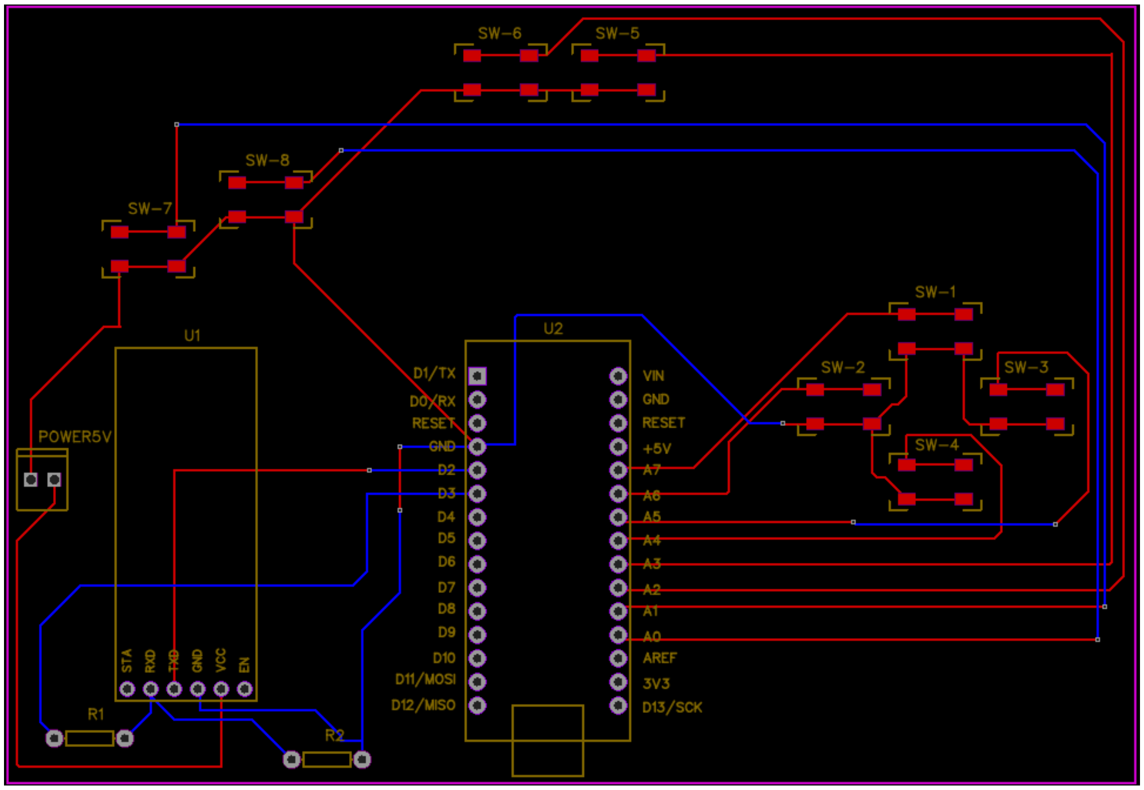


Figure 69 Designed Game Controller PCB

Each component represented in this PCB design, not including the power supply, will be surface mounted onto the PCB by the way of soldering.

4.2.12.2 Display Controller PCB Design

The main components of the display controller are the Teensy 3.2 and the 74HCT245. In order to simplify the design of the connections between the Teensy 3.2, 74HCT245, and RJ45s, a schematic was designed so that a small-scale PCB could be implemented. The Teensy 3.2 is connected via through holes to the PCB. The PCB is organized so that the USB ports fit the physical layout of the K.I.T. The Teensy outputs are routed through the PCB to the 74HCT245 buffer inputs as shown in Figure 70. The 74HCT245 outputs are routed through the PCB to the RJ45 jacks. The Teensy 3.2 is used to control the LEDs on the game board. The buffer will step up the voltage signal from the Teensy and provide the LEDs with the appropriate voltage signal so that every LED on the board will work properly. These designs were all made using EasyEDA to be consistent with the design process. The connections between all components are displayed in the designed schematic in figure 69 below.

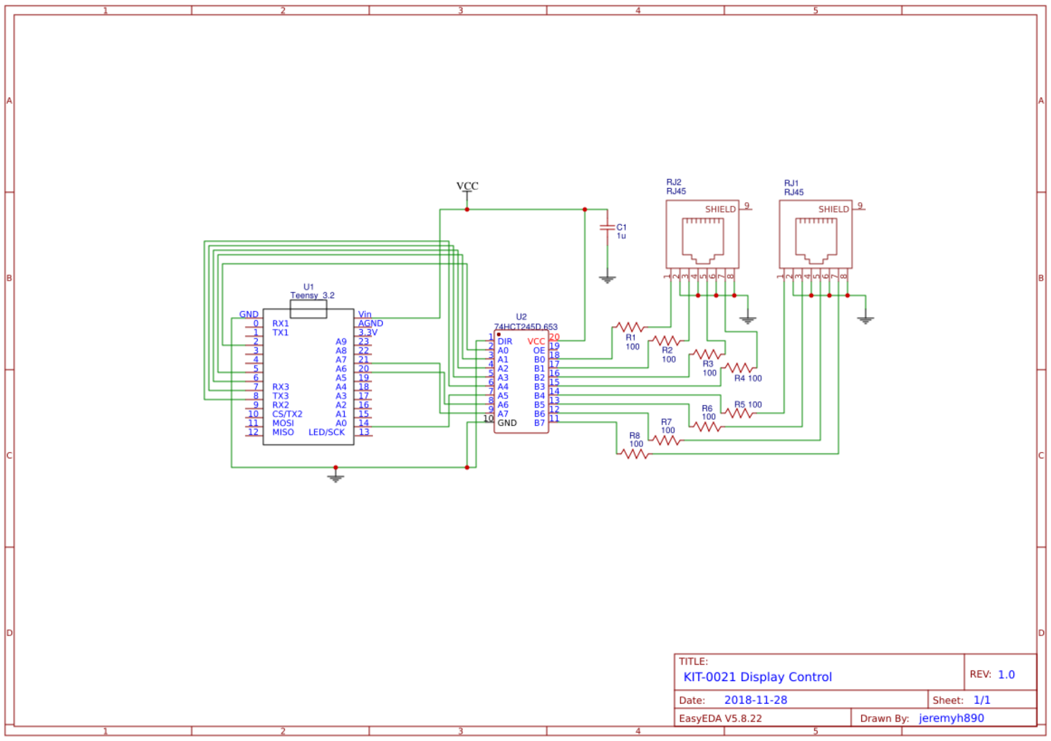


Figure 70 Display Controller Wire Schematic

After designing the schematic, it was exported to a PCB and the required connections were made between each component. Each of the components on the PCB will be surface mounted. The designed PCB is displayed in figure 71 below.

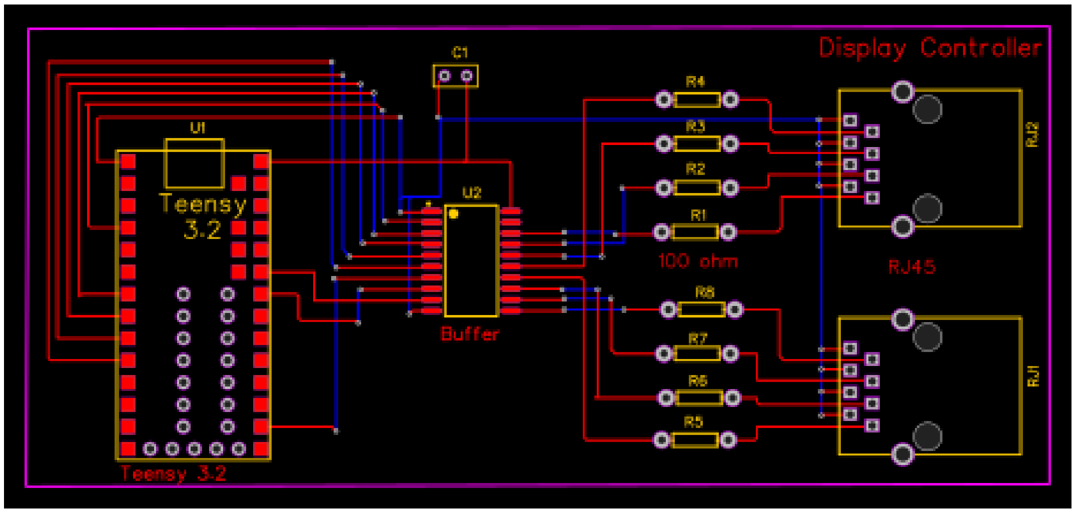


Figure 71 Designed Display Controller PCB

4.2.13 Component Verification

Some testing was done for the major components purchased. The following sections entail the verification of each component tested along with pictures for proofs that the components properly function. The schematics for each test will be included in a later chapter that includes the whole system tests. Each major component of the game controller is verified by ensuring that they power on when applying the correct voltage as instructed by each components data sheet. The power supply and power supply switch were verified using a DC power supply.

4.2.13.1 Handheld Controller Component Verification

Figure 72 below displays the verification of the Game Controller components. The standalone left image shows the Raspberry Pi 2 and Teensy 3.2 being tested with the red LEDs on both showing that they are working properly. The top left image shows the Arduino Nano being tested with the red LED indicating that the Nano is working. The top right image shows the HC-05 being tested with the red LED indicating that the HC-05 is working. The bottom left image shows the connections between the Arduino Nano and the push buttons. The bottom right image shows the connection between the HC-05 and the Arduino Nano with the red LEDs indicating that both modules are working properly.

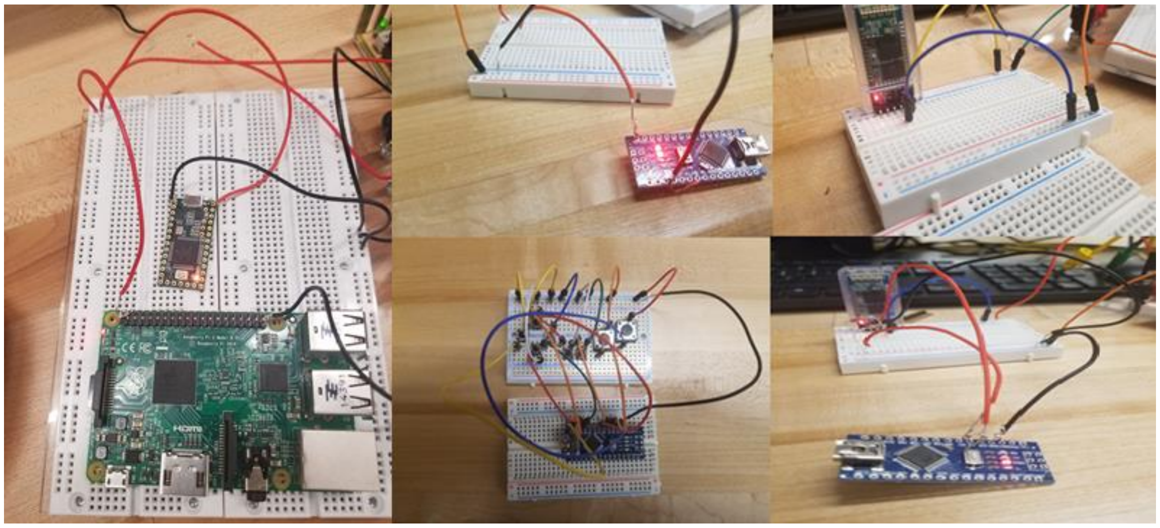


Figure 72 Component Verification Images

All of the test circuits were built using Elegoo breadboards, wires, and a Keithley 2230-30-1 Triple Channel DC Power Supply. Wires were coiled around input terminals when needed in order to establish the necessary connections. As each of these components powered on correctly, they are verified to be working.

4.2.13.2 Power ON/OFF Switch Test

After the switch has been successfully assembled to the three-prong power cable, we must perform a test to make sure that the power cable still functions correctly and to ensure that the connection between the screw terminals within the switch is tightened. To test that the cable still works like it is intended to, the female side of the cable is cutoff and strip to exposed some of the copper strands. The strands are then connected to the multimeter via two leads; one connecting to the live (black) wire and the other connected to ground and neutral (green and white) wires. Figure 73 below displays the power supply and switch test.

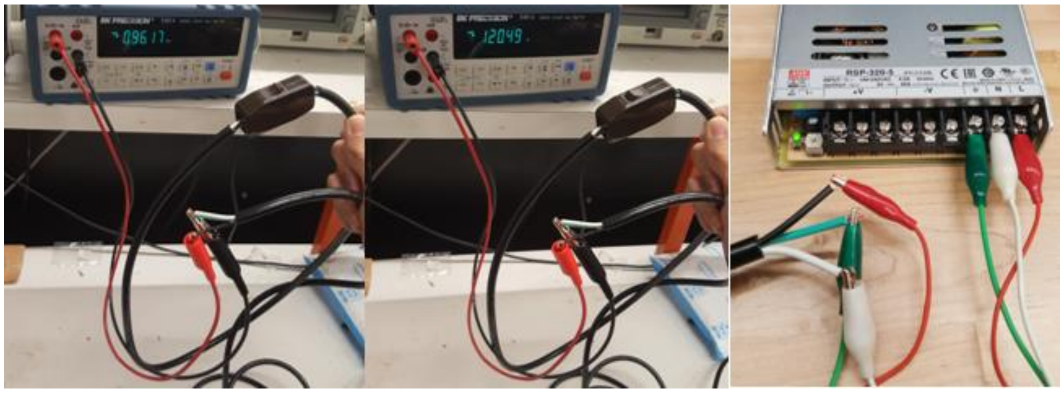


Figure 73 Power Supply and Switch Test

After all the wires are hooked up and is doubled check, the male part of the power cable is connected to an AC outlet. The expected voltage that is to be read through the multimeter is 0 volt since the switch is opened, not letting any current to flow from one terminal to the other. The left picture in Figure 73 shows the switch in its OFF position or an opened switch, letting no current through.

The switch is toggled to an OFF position or a closed switch. Now, the current is flowing through the switch from the AC outlet to the multimeter as shown in the middle picture in Figure 73. The multimeter reads roughly 120 volts AC, which is the rating for AC outlets in the United States. It can be concluded that the switch is assembled correctly and that the power cable functions as it should and is safe to use on a power supply any other operations. Since the switch did not come with marked ON/OFF position, we will have a small mark to indicate that the switch is in the ON position.

After the power cable is tested for its functionality and is determined that it works perfectly and is safe to use, it is used to power up the power supply to perform tests on the power supply. Some alligator clips are used instead of the 16 Gauge wires because the terminal rings that were bought do not fit into the small slots between all the terminals of the power supply. The correct terminal rings are ordered and shipped.

At this moment, the power cable should be unplugged with the switch in the OFF position, although it should not matter since the cable is not plugged either way. For this test, the end of the three-prong power cord is connected in series to the ground, neutral, and live terminals of the power supply as shown in the right picture in Figure 73, alligator clips are used in this case.

After all the connections are made and secured, the cable is plugged in to the AC outlet and the switch is toggled closed to an ON position. The LED indicator on the power supply is lit up to indicate that the power supply is ON, hence it is getting the power necessary to turn it on from the AC outlet. We could also hear the internal fan running when the power is on.

To test that the power supply outputs the correct output voltage, a multimeter is used to measure the output voltage between one of the positive terminals and one of the negative terminals. The expected output voltage is 5-volt DC.

The first reading was slightly higher than 5 volts. There is an adjustable potentiometer on the power supply that is used to adjust the desired output voltage. It is adjusted until the output voltage of exactly 5 volts DC is obtained. After it is sure that the output voltage coming out of the power supply is exactly 5 volts, the LEDs strips can then be connected to the power supply and do further tests with LEDs strips in order to figure out the voltage drop.

4.2.13.3 LED Strip Verification Test

Testing the compatibility with the power source was performed by connecting three 300 LED strips to the power source as shown below. With all LEDs turned to full white visually inspect the cable of changes in color. This test yielded that the power source could not handle powering 900 LEDs without signal loss. This is indicated by the change in color toward the end of the strip. However, in the final design 3 strips of 96 LEDs will need to be powered per terminal on the power source. A test was performed to see if 3 strips of 96 LEDs connected to one terminal could be powered to full white. The result indicates that the power source is capable of powering 288 LEDs through this connection as shown in the figure below. Because this was not a full test of all 768 LEDs intended for the design it is difficult to say that the power source will supply enough power. From the test of powering 900 LEDs failure it is decided to consider a different power supply or multiple power supplies for the design. Figure 74 below displays the LED strip verification tests.

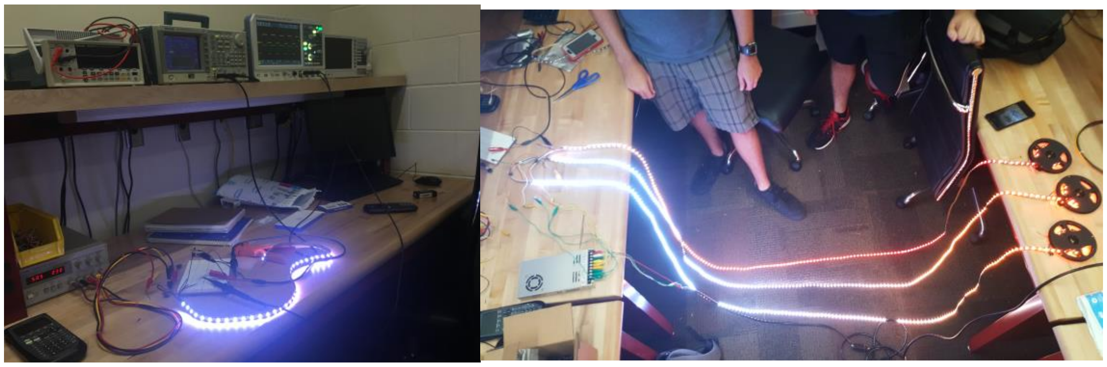


Figure 74 Testing Power Consumption of LED

4.3 Software

This section details all the software related components of the project ranging from game logic to GUI design. For K.I.T. our group decided to produce three games to show off what the system could do and provide a demo for the uses of our system. The team decided that the three games should be a matching game, a pong variant, and a dungeon game. These games would show off the system and would draw in potential users of the system with examples of the types of games they could run on the system or code for the system. The games were chosen with specific reasons in mind. The matching game was chosen to show that small simple games could be easily programmed with the system to draw in users that wanted to practice coding with the K.I.T. The pong variant game was chosen to show off the system's multiplayer capabilities with a classic game design. The dungeon game was chosen to show off what a more complicated game could be like on our system.

Many other games were discussed and ultimately these three were chosen. Other games could be added later in the life cycle of the project, but these three games will be the main focus of the project. Even with a simple LED display, the wide range and size of games that can be offered by the K.I.T. is immense. The limiting factors are only the size of the game, the display, and the programmer's imagination.

Currently, the K.I.T. will be constructed using a Raspberry Pi 2 to be the main microprocessor for game logic [61]. Raspberry Pi will allow the use of the Python programming language for our programs. Python is easy to program with and will increase the accessibility of the K.I.T. by using a popular language. The Raspberry Pi offers excellent processor speed for games to run on.

Another option considered was the use of an Arduino Uno for the main processor for the K.I.T. The Arduino currently has fewer libraries to help in programming, and we would have sacrificed the Raspberry Pi's utility and processor speed [62]. The Arduino would also have been much cheaper in cost. The Arduino would have forced the games to be programmed in C++ to be properly compiled for the Arduino. The use of the Raspberry Pi also adds the ability to use a general system menu screen where programs and options could be selected. With an Arduino implementation, each game must be loaded individually and played on the K.I.T. With both implementations, a SD card must be used to hold the software and load from.

In development, the initial programming of the three games will be in python to get a sense of how to program each game and have a workable demo to test on. Once the code is workable, the python code will then be transferred onto the raspberry pi to ensure it can work in the K.I.T. Any python code libraries that will be used in development or for testing will need to be removed. During development the pygames library will be used as a quick and simple way to display a mockup of what the LED matrix might look like [63].

In the final release, the code will need to speak to pin outputs or another controller to tell the LEDs to turn on or off [64]. The main issue will be ensuring that the code will be usable and readable so that any transition is easy. Allowing the use of python allows the project greater flexibility since python is relatively simple to code and can bring more potential users since many people learn python and will be able to program their own games for our system.

The programming of each of the games would require that each LED is individually addressable, especially if individual LEDs are meant to change color. The games also require that the LED matrix be refreshable at an acceptable rate. The games will mostly draw the next frame on animation to a buffer and then send that information to the LEDs to change the actual LED Matrix. In order to increase refresh rate and minimize the time needed to refresh the screen, only the section of the screen that is being changed should be changed. The game logic also needs to support the controllers to allow for user input. Since most input is coming from controllers, an interrupt handling system to deal with user input seems to be the best option to ensure the games are responsive. Polling was also considered but passed in favor of the more responsive option. Next will be an in depth look at the game logic of each of the games to show how each game was chosen and published.

4.3.1 Game Logic

This section goes into detail on the implementation of each game type that can be played on the K.I.T.

4.3.1.1 Matching Game

The matching game was meant to be the simplest and most easily accessible game of the initial trio. For a simple game, there were several options debated. The game had to be single player, simple enough to be easily programmable as a demo, and be able to show off what the K.I.T. could do. A variant on a breakout game was talked about and discarded for potentially being too limited on the LED display. A snake game was also talked about, this would have been another good choice for our system as a basic game that could serve as a demo and an example to future programmers that wanted to develop their own software for the K.I.T. However, the matching game was chosen over the snake game to better show off the LED display. Matching games are relatively easy to program, and the scope of the game can be easily expanded. The snake game variant would make an excellent choice as an additional game that could be completed for the system if time permits.

The matching game itself basically consists of displaying a set of face cards that the user can chose from. The user would select two cards and see if the cards match. If the cards match, they are discarded. If the cards do not match, the cards are set back face down in the same position as they were before. The goal of the game is to match all of the cards until all the cards have been discarded. The matching game relies on the user memorizing the card’s position to correctly match all of the cards. This type of game is commonly called a memory matching game and is used to promote memory skills.

Programming this game for the K.I.T. would consist of creating an array of card objects. Each card object would be initialized to hold a reference to a sprite pattern and whether the card had been discarded or not. The main game should play out very simply. For each card in the array that was not discarded, the LED matrix would display a lit LED. The LED matrix could also be programmed to show a design for an un-flipped card, but more cards can be displayed by showing each card as a single LED. After all un-discarded cards are written to the LED Matrix, the user should be able to select from the shown cards. The card that the user has selected could be shown by blinking the LED that currently represents that card or by changing the color of the LED. If the user selects the card the sprite referenced by that card should be shown on the LED matrix.

Next, the LED matrix should be showing the cards again, except the card that was just flipped should be identifiable through blinking or color change. This process so far should represent the act of picking up a card and placing it back down. The next card that is picked by the user should be checked against the previously chosen card. If the card shares the same sprite reference, then the cards are a match. Both cards should be marked as discarded. When the user is next shown the LED matrix of cards, both cards should be represented by unlit LEDs to show that they have been discarded. The game continues until all cards are matched and discarded. The main loop should consist of drawing the array to the LED matrix, allowing the user to select a card, showing the card sprite, checking to see if the card matched the previous card, and then checking to see if the game is over. The game itself is very simple but shows off the LED matrix and the use of the controller. The program itself must contain the predefined sprites to ensure that they can be easily written to the LED matrix. The game should also be able to tell how many LEDs it has to work with when creating its display. Figure 75 displays the Matching Game class diagram.

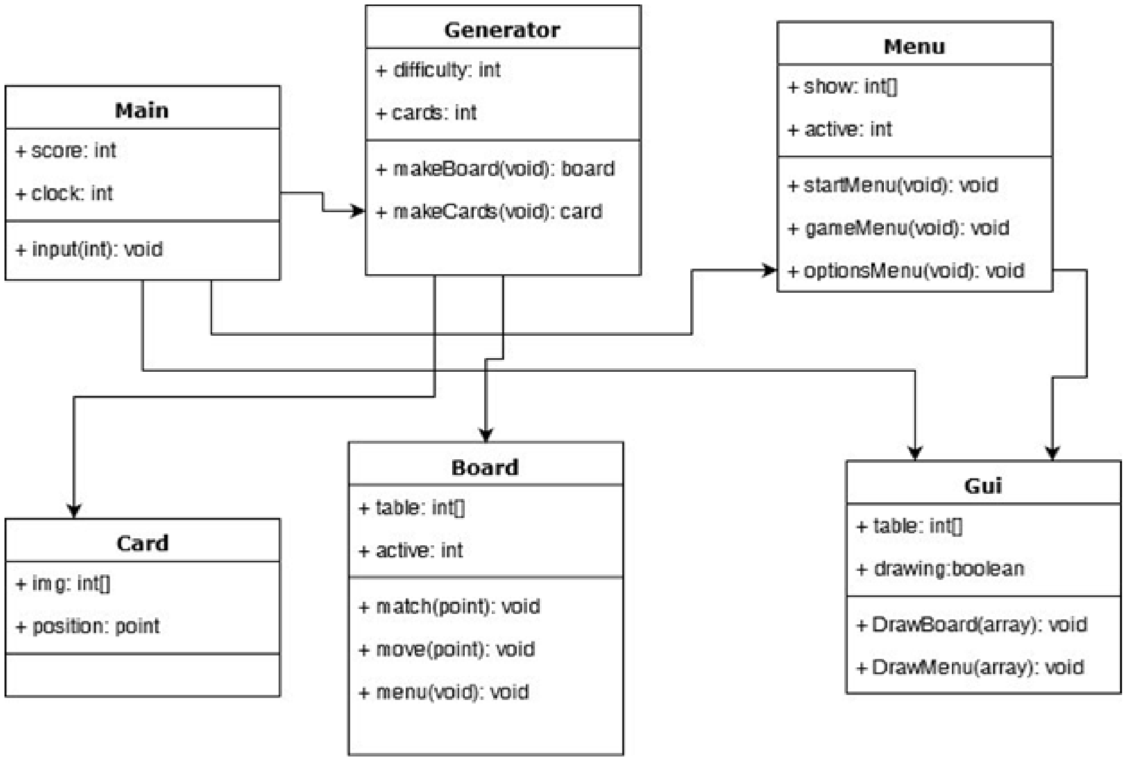


Figure 75 Matching Game Class Diagram

The matching game is an easy example of a program for the K.I.T. However, it can also be easily expandable. The game can be improved through the addition of more options such as a timer. The Matching game could easily time how fast the user completes the game and show it to the user to allow them to try to beat their old time. The game can also be increased in difficulty through the use of unmatched cards. There could be several cards on the board that do not have a match to any other card on the board. The game could identify these cards with a flag to ensure that when all other cards have been matched, the game ends. This would increase the difficulty when trying to complete the game if you keep trying to find a match to a card that has no match.

Another way to increase the difficulty of this game and add variation is to use randomly defined sprites. The game should always predefine the sprites it is going to use to allow for quick drawing to the LED matrix. However, the sprites being used could be randomly generated at the start of the game instead of pulled from a sprite bank that has more easily identifiable sprites. The randomly generated sprites would use a random number generator to create a unique sprite that fits in the display. Since it is randomly generated, the sprite would likely look very chaotic instead of a normal sprite of a pixilated symbol. This would force the user to pay even closer attention to the sprite designs and would greatly increase the difficulty. The addition of these options could be handled through an options menu.

4.3.1.2 Pong Variant Game

The pong variant game's main goal is to show off the multiplayer aspect of the K.I.T. When discussing options for a multiplayer type game for the K.I.T. several options were brought up. The first ideas discussed were board games such as checkers or Go. Both options would have been multiplayer, but pong was chosen over them since it was more dynamic and thought to be more interesting for the user. The next multiplayer game discussed was a side scroller multiplayer game such as a racing. This would have been an acceptable choice, but pong was chosen since it would be easier to program. If time permits, a side scrolling multiplayer game would be another excellent demo of the K.I.T.'s multiplayer abilities. Even though Pong was chosen since it would be easy to program and control, it was decided to add variations to the game to make it more interesting and more dynamic. These additions would be in the form of upgrades and power-ups that would randomly appear on the screen. The upgrades would make the player's board wider or cause more than one ball to appear on the screen.

The pong variant game would start with two player-controlled boards on either side of the LED matrix screen. The ball would start in the center of the screen and go in a random direction initially. The players would try to hit the ball with their boards before it touched their side of the screen. The players would also try to get the ball to touch the other player's side of the screen instead. If the ball touches the player's side of the screen, the other player would get a point, and the ball would restart in the center of the screen. The players would be able to control the side-to-side motion of their boards to try to deflect the ball to the other side. The power-ups would randomly generate in the middle of the field and would act on whoever last touched the ball. The game itself would end once a player had gotten a certain number of points. The scores of both players would be displayed on the top of the screen. Figure 76 below displays the pong variant class diagram.

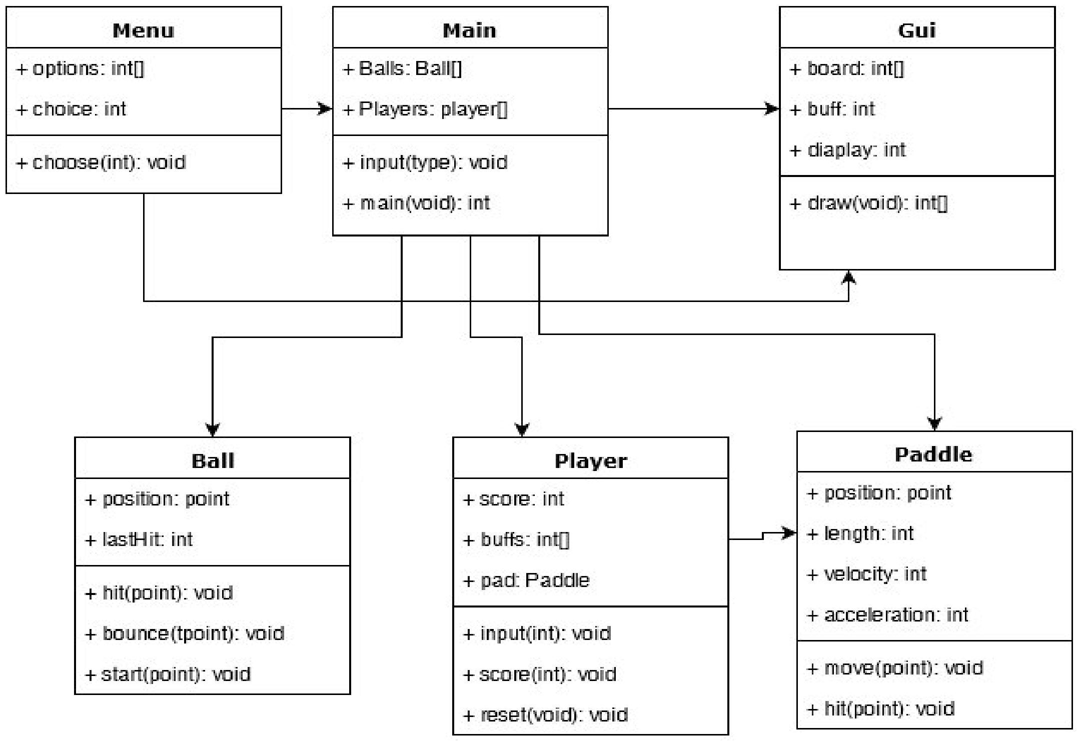


Figure 76 Pong Variant Class Diagram

The game would require objects for each player and the ball to hold their coordinates. The game would also need to be able to calculate the return trajectory for the ball once it hit something like a board or a wall. The game loop would consist of initializing the ball and players, checking the ball's speed and direction to see if there was a collision, moving the players and ball, and repeating until the ball hits a player's side. The logic defining the ball's behavior would be the most complicated part of the game. The ball would need to know if it had collided with something and what it was. The motion of the ball would also be important and would most likely require the ball to record values for X and Y speed and direction. The game would also require calculating a reflected trajectory based on where on the board it hit. The power-ups that might be implemented are adding additional balls, increasing the size of a player's board, and increasing the speed of the ball. In order to implement this, the ball object must also record the last player who touched it. Overall, the pong variant should be simple to program, but can easily show multiplayer and be expanded upon.

The Pong variant could easily be modified further to provide more options to the user if time on the project permits. The first easy modification would be to add more power-ups such as increasing the size of the ball, changing the speed at which the player can move. Other changes could be to increase the number of players from two to four to increase the multiplayer capability. Another possible addition to the game would be to add obstacles on the map to make the game board less open. The addition of these obstacles could be based on pre-defined maps or randomly generated. These options for additional gameplay could be controlled with an options menu or could simply be added to the game.

4.3.1.3 Dungeon Game

The dungeon game was chosen to be the more complicated game with a larger scope than the others. There were several other options to consider when choosing this type of large scope game. The first idea was an RPG type of game with a turn-based system. This idea was discarded as being too large and complicated to program on an Arduino. This game type could still be considered but was passed over in favor of a maze type game. The initial idea was a simple maze game where the player would start in a predefined maze and try to find the exit.

The game was expanded, and the dungeon game now has a completely randomly generated construction. The decision to make it randomly generated was made to increase replay ability. It was also decided to add multiple levels to the maze to increase complexity and scope. The maze also included a timer to push users to beat their old time and to create a sense of urgency. The game will also feature keys and other items to increase the options the user has when exploring the dungeon. The game itself is still a maze type game with the goal being to find the exit. However, additional complexity of items and random generation has made this simple design into more of a roguelike dungeon crawler.

The dungeon game's gameplay revolves around finding the exit to the mazelike dungeon. The levels and rooms are all randomly generated and can be radically different on every play though. The initial dungeon creation creates a predefined set number of levels. Each level is created by randomly generating a set of coordinates within predefined bounds. The coordinates are then connected to each other using an algorithm to generate each level's map. Each coordinate of the map is then made into a room. The rooms are randomly generated based on a set of bounds and doors are added to each room based on whether or not there is another coordinate next to the room's coordinate on this level. The levels are connected to each other through special doors that are randomly added to two coordinates on each level. One door goes to the level above and the other goes to the level below. The levels loop so that the last level will take you back to the first level if you try to go down again.

The door logic works very similarly with both doors from room to room and level to level. Each door is an object that holds the coordinates of the connecting door. When the player goes through one door they are moved to the coordinates of the other door. The player is able to hold up to three objects as they move through the dungeon, including keys. The dungeon will randomly generate keys and other items in each room on creation. The game screen will be displayed on the LED matrix. There will be multiple views on the LED matrix including a room view which will be centered on the player, a map view which will show the layout of the entire level, and an inventory view showing the items the player currently holds. Each view will be displayed at once on the LED matrix in different sections.

The game loop consists of drawing the state of the player to the LED matrix. The player's position in the room should be shown, but only to the limits of the space the room view has on the LED matrix screen. The coordinates that are too far away from the player would not be shown since they would fall outside the view window. The level map would act similarly, showing the player's position on the map. The player would be represented on the LED screen through a certain color of LED or by blinking the player's LED position. Walls and doors would be similarly shown on the LED matrix and would be shown as either distinct colors or a solid LED. The LED screen would redraw the room view every time the player moved. If the player moved thought a door the room view, and map view would need to be redrawn.

The item view would only need to be redrawn if an item was used or picked up. Only updating the parts of the LED screen that needs to be updated will cut down on the lag between action and result on the LED matrix and provide a smoother transition. Currently the dungeon game only ends when the player finds the exit which is another type of door. Once the game ends the player will be sent to a menu where they can choose to start another dungeon run or exit the game. Figure 77 below displays the Dungeon Game class diagram.

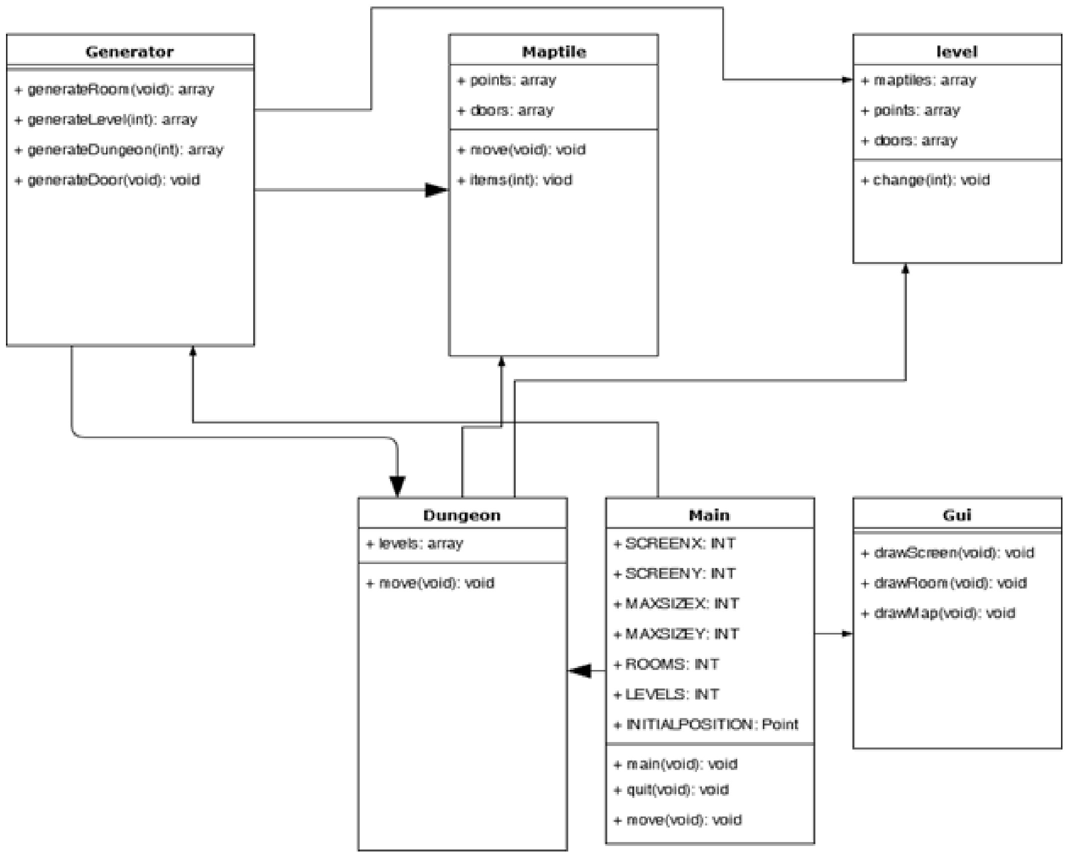
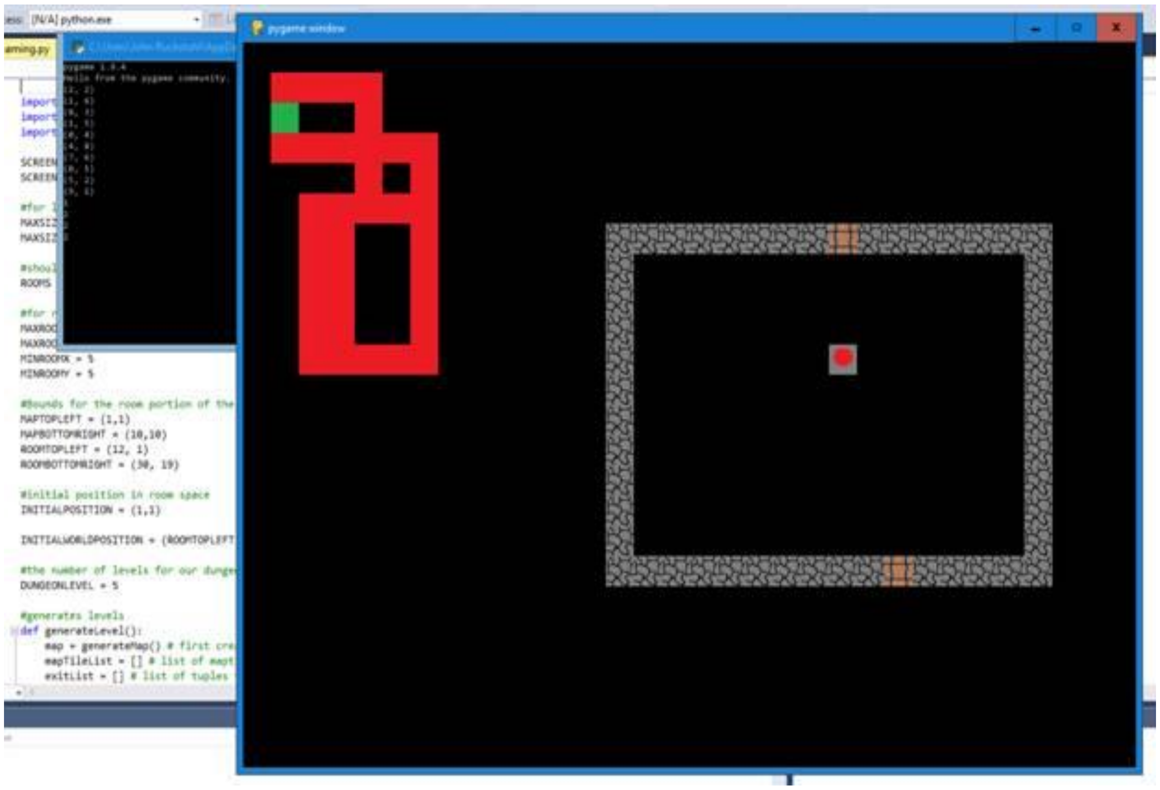


Figure 77 Class Diagram for Dungeon Game

The dungeon game has several possible modifications or additions that would increase the level of gameplay. The most obvious addition would be enemies. Adding enemies would be time consuming but would add an important adversarial element. Currently the only obstacle to the user's goal is time, adding enemies would add a more strategic element to the game. The constraints to adding enemies would be time to implement. Adding enemies would mean adding some sort of combat system or some other way to deal with the enemies. The other issue would be game balance to ensure that the game feels difficult without feeling impossible to the user. This might lead to the inclusion of weapons or armor into the game as items to help balance. While adding enemies is a goal for the game many other systems must also be added in addition to this. Another possible modification to the game would be the addition of pre-generated rooms.

Currently the game's levels and rooms are completely randomly generated. The addition of rooms that are pre-defined could add another level or gameplay. Many other randomly or procedurally generated games rely on a mix of random and predefined elements to ensure that their games are interesting. Adding pre-defined rooms to a randomly generated dungeon could come in the form of a boss room, a room with NPCs that you could talk to in it, or even puzzle rooms. These rooms would require extra work to ensure that they could fit in to the rest of the dungeon, but could add much more gameplay. Addition of pre-defined room into the dungeon would be another goal to add. Figure 78 displays an example of the dungeon game development.



```
pygame 1.9.4
Hello from the pygame community.
```

```
import
import
import

SCREEN
SCREEN

#for
MAXSIZ
MAXSIZ

#shoul
ROOMS

#for r
MAXROO
MAXROO
MINROOMX = 5
MINROOMY = 5

#Bounds for the room portion of the
MAPTOPLEFT = (1,1)
MAPBOTTOMRIGHT = (10,10)
ROOMTOPLEFT = (12, 1)
ROOMBOTTOMRIGHT = (30, 19)

#initial position in room space
INITIALPOSITION = (1,1)

INITIALWORLDPOSITION = (ROOMTOPLEFT

#the number of levels for our dunge
DUNGEONLEVEL = 5

#generates levels
def generateLevel():
    map = generateMap() # first cre
    mapTileList = [] # list of mapt
    exitList = [] # list of tuples
```

Figure 78 Dungeon Game Development

4.3.2 GUI

In addition to the LED matrix, the K.I.T. will also have a small LCD display on its side to show certain values. The debugging of most games will rely on this LCD screen to display error codes that can be evaluated by programmers. The LCD display will also show certain values like volume, which file is currently being run, and the connectivity of the controllers. The use of this LED screen will allow the LED matrix display to focus on gameplay rather than displaying options menus for several of the peripherals. Several options were discussed for the LCD display and the Monochrome 1.3" 128x64 OLED graphic display by Adafruit seems like our best option since it can easily integrate with an Arduino board and has libraries associated with it to make programming simpler [65].

The programming for the LCD should entail reading file names from the SD card, reading error codes from the running code, and taking readings from peripherals like the controllers and volume and displaying them. The GUI for the games discussed, will be displayed on the LED matrix when the game program has been loaded. The menu system should be simple with the name of the game, a game start button, and an options button if the game requires it. The simple menu style will allow the LED matrix to display the text without cluttering the screen or forcing users to scroll through a long menu with too many options. K.I.T. games should strive to have a very simple readable menu system for their GUI.

4.3.3 Game Controller Interface with Arduino App

The main microcontroller that runs the LED game board will have an HC-05 connected to it so that it can communicate via Bluetooth. In using the HC-05 Bluetooth module, it allows for the user to use their Android phone as a game controller. The app “Arduino Bluetooth Controller” allows the user to pair their phone to a Bluetooth device that is connected to an Arduino and choose a mode they would like to use to communicate with the Arduino. Table 26 below displays the game controller function.

Table 26 Game Controller Function

Mode	**Function**
Controller	Create a Game Controller interface to send commands to the Arduino
Switch	Control a remotely connected switch on the Arduino
Dimmer	Change values on your Arduino such as brightness
Terminal	Send custom commands and decode them on the Arduino side

In the case of this project, the Controller mode can be used to design a custom controller interface on the app that can send commands to the Arduino and play each of the games on the LED game board. The upside to using a phone as a game controller is that it is very low cost as you do not have to spend any extra money on building a physical Bluetooth game controller. Although an actual game controller is being built for this project, the app will be used as an extra option as a game controller. An example of this interface is displayed in figure 79 below.

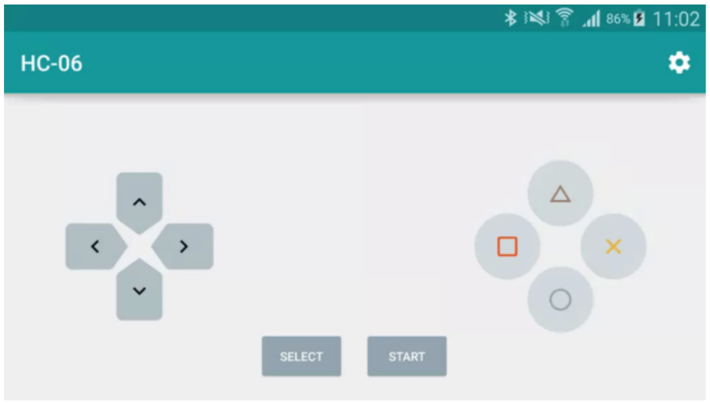


Figure 79 Arduino Bluetooth Game Controller App Interface [92]

4.3.4 Peripheral Component Code

For the K.I.T. project, it will be necessary to have communication between multiple microcontrollers with our current design. Some communications like the controller will be handled through Bluetooth protocol. For other microcontroller boards the communication protocol we have currently chosen is SPI. SPI will allow multiple boards to communicate with each other at a fairly high rate of speed, potentially 10 Mbps [94]. Both the Arduino and Raspberry Pi have SPI libraries to handle this communication format [95] [96]. The high communication speed will greatly benefit the project since reducing input lag and screen refresh rate will be important for quality control. Being able to reduce time communicating between boards will mean fewer bottlenecks that we will have to clear up during testing.

SPI does come with drawback such as requiring more pins to communicate through. This could turn into a problem later depending on if we require any more micro controllers. Adding on more micro controllers could require use to add more pins through a bus or switch communication formats. With our current design SPI makes sense, we have relatively few boards we need to communicate between, but we also need fast communication. The other communication format considered was I^2C. I^2C would have been a good option if we needed to connect more microcontrollers. It would have been able to communicate through fewer pins. This would have come at the disadvantage of only 1 Mbps of speed. If we need to add more micro controllers together, this might be another option.

Another important aspect of coding for the peripherals is their use as an embedded system. For the Arduino development boards, this is not an issue since the Arduino will simply use its bootloader to run whatever program it has loaded. For the Raspberry Pi, some additional work will be needed to auto run whatever programs the project will need. In order to do this the Pi will need to be set up to auto launch a program on startup. This program will need to not only be able to run functions like playing the game software, but will also need to establish connections to the other micro controllers. Fortunately, the raspberry Pi has several options to allow it to run programs on startup [97]. This will allow a startup sequence to be run through the Raspberry Pi without any additional devices like a PC.

One goal for the project is to be able to have the K.I.T. be completely standalone without any external source helping it, aside from a power cable. The start-up program will need to run a sanity check on the K.I.T. ensuring that all components are correctly communicating with the Pi and each other. The startup sequence should also be able to communicate start-up errors to the user. The mini LCD screen will be used to handle any error messages for the K.I.T. The start-up sequence should also be able to complete and move the player to the main function of the K.I.T. through a menu screen. The initial startup code will be just as important to the project as the game software. Without being able to properly work as an embedded system the Raspberry Pi would not be a useful to the project and another micro controller would have to be used.

4.3.5 System Menu

Another important software function to develop will be the system menu. After the startup sequence, the K.I.T. should show the user a menu system. Several option functions for the K.I.T. will be handled though user input with buttons or knobs, such as the brightness dimmer or volume. However, other options will be handled through the use of a menu system. The user should be able to use the menu screen to select which game they want to play. The Raspberry Pi will need to be able to read what game files it has and display them to the user. This will be handled through the menu system code.

The system menu should also handle option selection for controllers, color selection, and system information. Any additional configuration information or selection functionality should be added to the system menu rather than in individual game files. Most game produced for the K.I.T., including the ones that were described above, will come with their own menu screen. However, their menu functionality should only pertain to that individual game rather than the system itself. The system menu should be run by the Raspberry Pi since it will be required to read files from its microSD.

The main display for the system menu should be the LED board. The LED board should have enough space to display the menu items and any additional options. However, the main issue with displaying information through the LED board will be its screen resolution. The menu system will need to show multiple lines of text. It could be difficult to render that much text with letters and numbers that are readable. In order to help with this, a graphical representation guide for ASCII text for the KITT was designed. This representation used letters and numbers that could be rendered using four LEDs across and five LEDs down. Using this guide, four rows of text with eight letters each could be displayed on our 24x32 LED board. Despite this, the menu system will likely require multiple pages and a scrolling feature for any menu options or titles that are too long to be completely rendered on one screen. In order to follow standard conventions, the design philosophy for our menus will be to keep the menus as simple and consistent as possible for the user [98]. Figure 80 below displays letter and number examples.

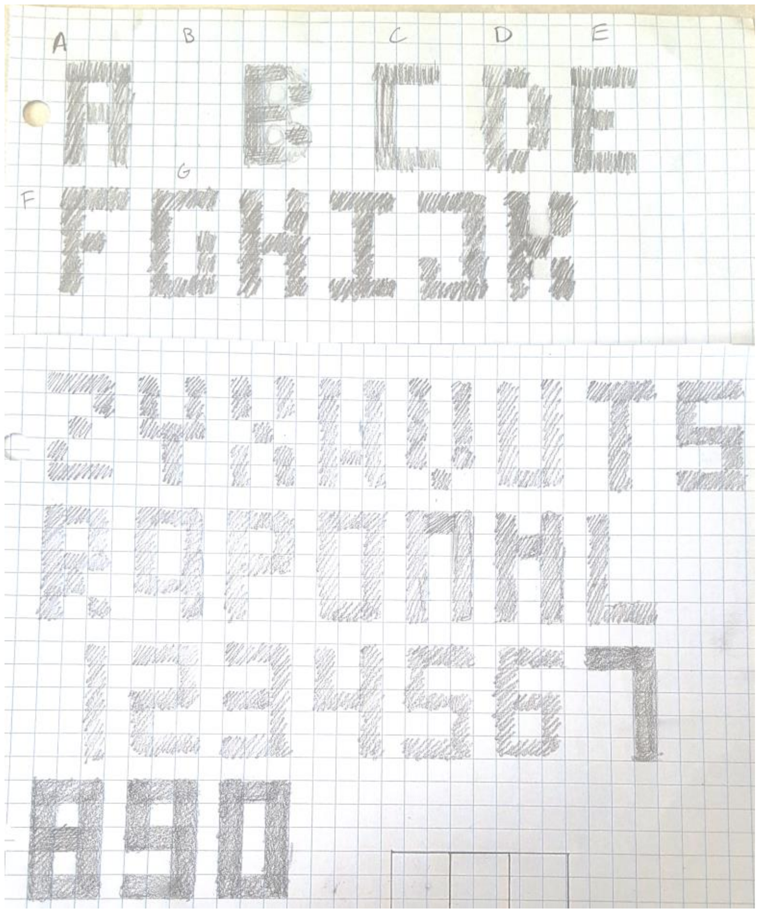


Figure 80 Example of System Letter and Number values A-Z, 0-9

The system menu should be something that most users don't notice and don't spend a lot of time on, but it will still be necessary for the function of the system. Our system menu should be able to be used by future users and content creators or the K.I.T. To keep that in mind, the files for the system menu will be available to the user in order to let the user customize the menu system however they see fit. Since the K.I.T. is meant to be user friendly and for user to develop their own content for, allowing users access to these files seems in line with the K.I.T.'s goals.

4.4 Reference Design

This section is dedicated to the design information that was utilized for understanding how each subsystem operates. From diagrams of processors used in the subsystem to schematics of the development boards this section is to better understand all the components that go into making the K.I.T. function.

4.4.1 MK20DX256VLH7 Processor

This MK20SX256VLH7 Processor is the backbone to the Teensy 3.2. Understanding this processor is vital to understanding how the Teensy 3.2 operates. Shown below in figure 81 is the pinout of this processor [57].

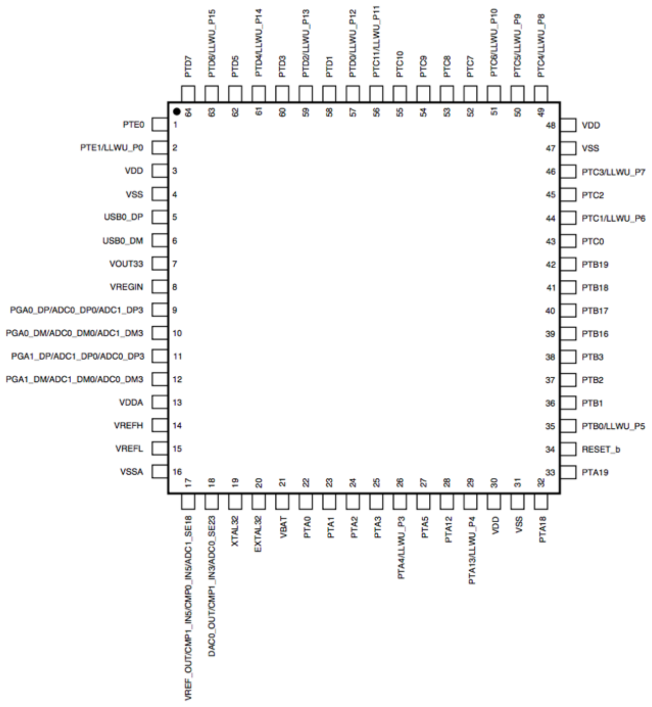


Figure 81 DX256VLH7 Pinout Diagram

The processor characteristics are shown in the tables below. In table 27 the power characteristics are shown. This information is needed to understand the power constraints of the Teensy 3.2 processing capabilities [57].

Table 27 Voltage and Current Operating Ratings

Description	Min	Max	Unit
Digital Supply Voltage (V_{DD})	-0.3	3.8	V
Digital Supply Current (I_{DD})	-	185	mA
Digital Input Voltage (V_{DIO})	-0.3	5.5	V
Analog, RESET, EXTAL, XTAL input voltage	-0.3	VDD + 0.3	V
Maximum Current Single Pin	-25	25	mA
Analog Supply Voltage	VDD - 0.3	VDD + 0.3	V
USB-DP Input Voltage	-0.3	3.68	V
USB_DM Input Voltage	-0.3	3.68	V
USB Regulator Input	-0.3	6.0	V
RTC Battery Supply Voltage	-0.3	3.8	V

In table 28 the environmental electric characteristics are displayed. This information is used to understand the environment and handling characteristics of the Processor to ensure failure does not occur during the integrating and installation procedures [57].

Table 28 Electro-Static Discharge Ratings

Description	Min	Max	Unit
Electro-Static Discharge (CDM)	-500	500	V
Electro-Static Discharge (HBM)	-2000	2000	V
Latch-up Current	-100	100	mA

Table 29 displays environmental handling characteristics. This information is used to understand the environment at which the processor is used as well as the temperatures it can be stored at [57].

Table 29 Handling Ratings

Description	Min	Max	Unit
Moisture Sensitivity Level	-	3	-
Storage Temperature	-55	150	C
Solder Temperature	-	260	C

4.4.2 74HCT254 Buffer Chip Characteristics

This buffer is used to amplify the 3-volt output signal of the Teensy 3.2 to a 5-volt signal needed by the LED strip. The following information is used for reference to determine the understand the capabilities of this buffer. Figure 82 below is used for reference in creating the PCB design [58].

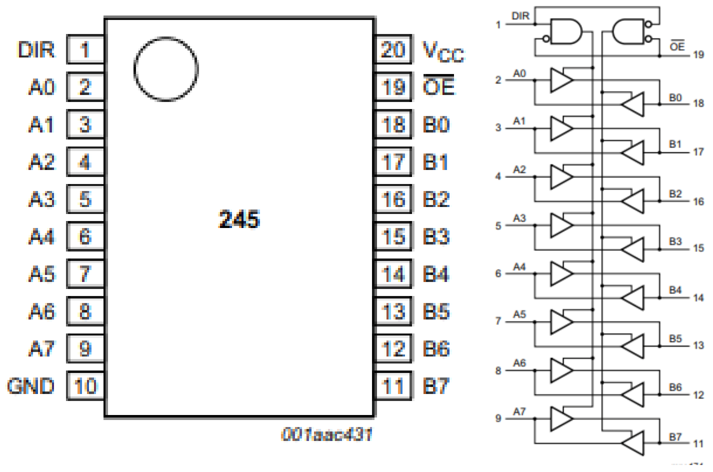


Figure 82 74HC245 Pin and Logic Diagram

The Buffer/Driver ratings and characteristics are shown in the tables below. In table 30 the power ratings are displayed. This information is used to understand the delay added by utilizing a buffer, the power required to operate the device and the temperate range of operation [58].

Table 30 74HCT254 Recommended Operation

Parameter	Recommended	Max
Supply Voltage	5 V	5.5 V
Input Voltage	-	Vcc
Output Voltage	-	Vcc
Input Transition Time	1.67 ns/V	139 ns/V
Temperature	25 C	125 C

Table 31 displays the static electric characteristics which is used to understand what voltage range the buffer can accept [58].

Table 31 74HCT245 Static Characteristics

Parameter	Typical	Max
High Input Voltage	1.6 V	5 V
Low Input Voltage	0.8 V	1.2 V
High Output Voltage	4.5 V	-
Low Output Voltage	0	0.1 V
Power Supply Current	-	8 uA

Table 32 displays more information regarding delay and time characteristics of the buffer. This is necessary to ensure the buffer does not increase delay time too significantly [58].

Table 32 74HCT245 Dynamic Characteristics

Parameter	Conditions	Max
Propagation Delay	An to Bn	33 ns
Enable Time	OE to An/Bn	45 ns
Disable Time	OE to An/Bn	45 ns
Transition Time	VCC = 4.5	18 ns
Power Dissipation Capacitance	per buffer	30 pF

5. Project Testing and Prototype Construction

The following section details the testing of hardware and software components for the K.I.T. design and the results associated with the testing. The sections are broken down into LED Display, Audio, Game Controllers, Power supply, along with gaming software testing.

5.1 LED Display Overview

This section details the testing involved with ensuring the LED display is operation. The LED display is operated using the TEENSY 3.2 development board and the WS2812B LED strip. The testing is separated into two sections, testing the WS2812 LED, testing the TEENSY 3.2 microcontroller, and testing the Buffer/Driver to boost the 3.3-volt signal from the Teensy to a 5-volt signal required by the WS2812 LED strip. At this stage, the LEDs are being powered using a power supply.

5.1.1 Testing WS2812B LED Strip

Testing the LED display starts with determining the requirements associated with the LED Display. The main requirements associated with the display are the number of LEDs and the resolution of video capabilities. The project requires a minimum of 768 LEDs for display and a resolution of 24 x 32. Testing that these requirements are met is straightforward. First verify the resolution of 24 x 32 by counting the number of rows of LEDs and then the number of columns of LEDs. The number of rows is equal to 24 and the number of columns is equal to 32. To verify that the display utilizes 768 working LEDs ensure the display is on. Alternate each LEDs color using the display controller to easily determine each operation of individual LEDs.

With each row displaying 32 operating LEDs there are 768 operation LEDs. Testing the display goes further than requirements however in that the display must have multicolor function for game display. Using the display controller set each LED component to full RED. Validate each LED is full RED. Using the display controller set each LED component to full GREEN. Validate each LED is full green. Using the display controller set each LED component to full BLUE. Validate each LED is full BLUE. Because the variation of colors is created by combination of RGB LEDs this test validates that each LED component is capable of color variation.

Testing the LED power consumption and power source compatibility is also important to ensure a functioning display. To test the power consumption first, a strip of 52 LEDs is connected to power and turned on to full. This is because at full white the LED strip will consume its maximum amount of power. With a multi-meter between the power source and the LED strip the current is measured to be 2.3 amps or 0.044 amps/LED. This indicates the LED strip consumes 0.22 Watts per LED.

5.1.2 Testing Teensy 3.2 Development Board

There are no direct requirements outlined for the Teensy 3.2 development board. That said the Teensy 3.2 must be capable of producing the proper signal timing to update the addressable LEDs. The Teensy 3.2 must also be capable of receiving information via USB to be compatible with the raspberry pi.

Testing the LED communication capabilities of the Teensy 3.2 begins by powering on the Teensy 3.2. Each output pin dedicated to driving the display must be tested. Start by connecting an oscilloscope lead to pin 2 of the Teensy 3.2. Program the Teensy 3.2 to output the necessary signal to display green for 96 LEDs.

As the Teensy outputs the necessary signal for the WS2812B internal circuit measure the period of each bit. Refer to the WS2812B design section to understand what signal is required to display green for an individual LED. Measure the period in which the output is high or 3.3 volts and compare this to the period in which the output is low or 0 volts. The first 8 periods should indicate high for approximately 800 nanoseconds and low for 450 nanoseconds. The following 16 periods should indicate high for approximately 450 nanoseconds and low for approximately 800 nanoseconds. This corresponds to a bit pattern of 111111110000000000000000 in binary.

In hexadecimal this patter is FF0000. Referring to the WS2812B datasheet this signal will produce result in each LED display full green. Repeat this process for Pins 14, 7, 8, 6, 20, 21, and 5. Small variation in the timing is acceptable. Refer to the design section for the high bit, low bit, and bit period tolerances of the WS2812B. To validate that the LED display is operating nearly simultaneously connected an oscilloscope lead to all 8 pins. Because of the limited oscilloscope inputs this test was performed by connecting a lead to pin 2, 7, 6, and 21. Compare the timing differences between each signal.

The idea behind this test is to see if the first set of 96 LEDs receive the display information and the last set of 96 LEDs receive display information fast than the user can perceive a difference. The human eye operates at approximately 32 frames per second. If it takes more than 0.32 seconds from the first LED to be altered and the last LED to be altered the user can theoretically perceive the delay. This can be approximated by seeing how long it takes for 96 LEDs to be updated.

Then determining the difference in time between when the first LED of the first 96 set is updated to when the first LED of the last LED set is updated. By adding the amount of time, it takes to update 96 LEDs to this difference in time between the first set and last set it can be determined how long it takes to refresh the entire display. The K.I.T. is designed to update the display at a minimum of 1/30th of a second.

5.1.3 Testing the 74HCT245 Buffer

Finally, to complete the display testing it is necessary to test the 74HCT245 buffer/driver chip. This chip is design to amplify the 3.3-volt output signal from the Teensy 3.2 to the necessary 5 input signal for the WS2812B LED strip. To perform this test, connect a 3.3-volt square wave frequency of 2.2 MHz to the input pins of the 74HCT245. Measure the output pins using an oscilloscope to determine that the frequency is still 2.2 MHz and that the new output voltage is 5 volts.

5.2 Arduino Nano Test Plan

In order to test the Arduino Nano to ensure that it is working properly, a basic circuit should be made on a breadboard that will give the microcontroller the correct amount of power. To test this component, an Elegoo breadboard, DC Power supply, and wires will be used. Connect the attachable terminals that come with the Arduino Nano to the breadboard and mount the Nano onto the terminals. Connect a wire to the power terminal on the Nano and to the ground terminal. Using the DC Power supply, ground the negative power rail on the breadboard and apply a 5V voltage to the positive rail on the breadboard. Connect a wire from the positive rail to the power terminal of the Nano. Connect a wire from the negative rail to the ground terminal of the Nano. Turn the power of the DC supply on and make sure that the Nano's LED turns on indicating that it is working properly. The test schematic is displayed in Figure 83 below.

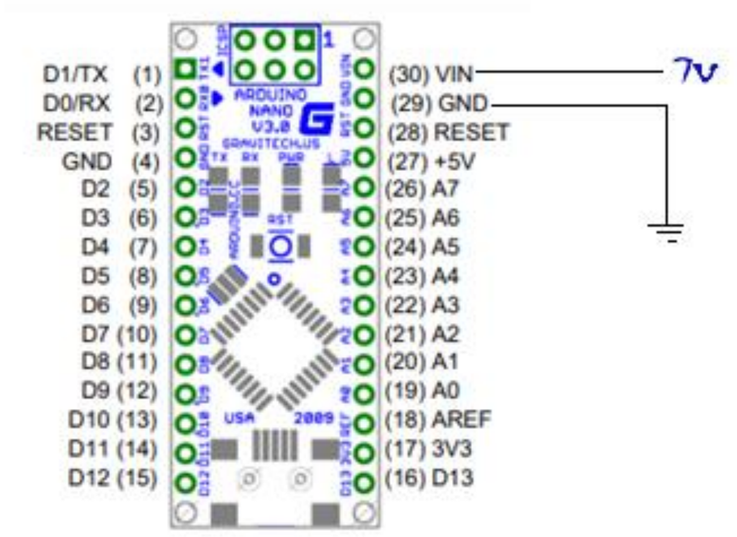


Figure 83 Arduino Nano Test Schematic

After verification that the Nano is powering properly, the Nano must be tested to ensure that all functions of the Nano are working properly. To test its functions, the Arduino IDE will be used. The Nano should be plugged into a PC via USB. A simple program will be written that will blink an LED on the Arduino Nano. Once the program is uploaded into the Nano and ran, if the LED does blink as expected then the Nano is functioning properly.

5.3 HC-05 Test Plan

In order to test the HC-05 to ensure that it is working properly, a basic circuit should be made on a breadboard that will give the Bluetooth Module the correct amount of power. To test this component, an Elegoo breadboard, DC Power supply, and wires will be used. Connect the terminals of the HC-05 directly into the breadboard ensuring that each terminal is connected to a different node. Connect a wire to the power terminal on the HC-05 and to the ground terminal. Using the DC Power supply, ground the negative power rail on the breadboard and apply a 5V voltage to the positive rail on the breadboard. Connect a wire from the positive rail to the power terminal of the HC-05. Connect a wire from the negative rail to the ground terminal of the HC-05. Turn the power of the DC supply on and make sure that the HC-05's LED turns on indicating that it is working properly. The test schematic is displayed below in Figure 84.

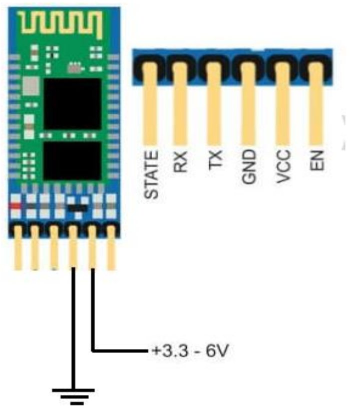


Figure 84 HC-05 Test Schematic

Once the HC-05 is verified to being power on properly, its Bluetooth function will be tested. In order to test the HC-05, it should be paired to a computer that has the HyperTerminal software installed. This software enables the user to interface with devices. Using the HyperTerminal software, AT commands will be sent to the HC-05. If the HC-05 returns with "OK" then the HC-05 is working appropriately.

5.4 Arduino Nano Connection to HC-05 Test Plan

In order to test the connection between the Arduino Nano and HC-05, a basic circuit should be made on a breadboard that will give the microcontroller and Bluetooth Module the correct amount of power. To test this component, an Elegoo breadboard, DC Power supply, 1k resistor, 2k resistor and wires will be used. Using the DC Power supply, ground the negative power rail on the breadboard and apply a 5V voltage to the positive rail on the breadboard. Connect the terminals of the HC-05 directly into the breadboard ensuring that each terminal is connected to a different node. Connect the attachable terminals that come with the Arduino Nano to the breadboard and mount the Nano onto the terminal ensuring that no connections have been made to the HC-05 yet. Connect a wire from the transmitter terminal of the HC-05 to terminal D2 of the Nano.

Using a wire connect terminal D3 of the Nano to a 1k resistor. Connect a wire from the ground terminal of the HC-05 to a 2k resistor while connecting the 2k resistor to a node shared by the 1k resistor. Connect a wire from the HC-05 receiver terminal to the node shared by both resistors. Ground the 2k resistor and the ground terminal of the Nano using wires connected from the negative power rail. Apply the 5V voltage to the power terminal of the HC-05 using a wire connected from the positive power rail. Turn the power of the DC supply on and make sure that the Nano's and HC-05's LEDs turn on indicating that they are working properly. The test schematic is displayed below in Figure 85.

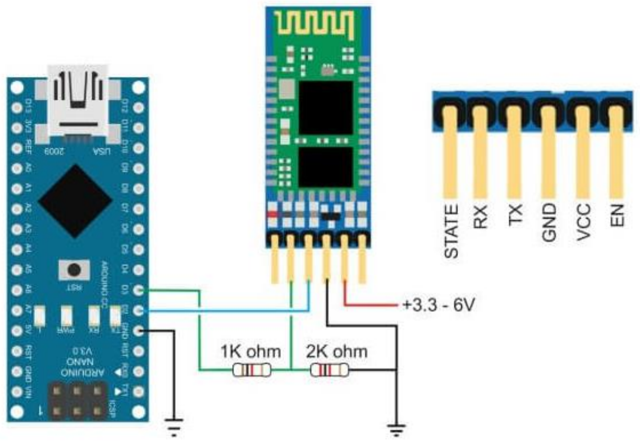


Figure 85 Nano to HC-05 Schematic [66]

After testing to make sure both devices power on using the above schematic, their functionalities must be tested. To test this, the HC-05 will first be paired to a computer. Once paired, a program will be written on the Arduino IDE that will blink an LED on the Arduino Nano. This program should be uploaded into the Arduino Nano using the HC-05 Bluetooth module. Once the program is running, the Arduino Nano should blink exactly as expected. The goal of this test is to ensure that the Nano has the ability to transmit and receive data via Bluetooth which it does not initially have the capability to do.

5.5 Game Controller Button Layout Schematic

The schematic for the game controller was first prototyped on a half-sized Adafruit Perma-Proto Solderless Breadboard. This breadboard allows for a simple transition to a custom-made PCB if needed. In order to create the schematic, six 6mm x 6mm x 12mm tactile buttons and two 3mm x 6mm x 2.5mm tactile buttons were needed. The button layout for this game controller was drafted in an earlier section. The two smaller tactile buttons were connected at the bottom of the breadboard. These buttons will serve different functions depending on the game being played. Two of the 6mm x 6mm x 12mm were connected on the right side of the breadboard. These buttons will act as selection buttons. The remaining four tactile buttons were connected on the left side of the breadboard. These buttons will serve as the directional pad. The buttons were then connected to different input terminals of the Arduino Nano. The overall button layout schematic is displayed in the component verification section.

With the button layout designed on a breadboard, the input buttons on the Nano can be tested. Using the Arduino IDE, a program will be written that assigns each tactile button to an input pin on the Arduino Nano. Each button will be programmed to send a 1 to a HyperTerminal window on press to show that they are working. Once the program is uploaded to the Nano, every button press should display a subsequent 1 on the HyperTerminal window.

5.6 Game Controller Test Plan

In order to ensure that the game controller is working properly, it was tested multiple ways. This test plan assumes that the controller's buttons are programmed as keyboard keystrokes. By connecting the controller via USB to a computer, the computer should recognize the device. By opening a blank word document and pressing buttons on the controller, the keystrokes that were programmed for each button should appear on the word document as if you were typing on a keyboard. To further test the controller, a connection via Bluetooth should be made to the computer. The computer should recognize the HC-05 device that is connected to the Arduino Nano and enable the user to pair the devices. Once paired, the user should be able to use a blank word document to test whether or not the proper keystrokes are input on the word document.

Once the game controller is fully functional, it should be programmed for the specific games that will be played on the KIT. For each game, the game controller should be examined to make sure that the controller is sending the correct inputs in reference to each game. For example, when playing the matching game on the KIT, pressing the selection button on a card should flip that card over and display the card's image. Using the directional pad buttons should allow you to hover over different cards. Using a square button should take you to a game selection menu. If the game controller performs these tasks correctly then the test is complete for the specific game being played.

5.7 Power Related Tests

To ensure the safety of everyone in the team and users, tests should be performed for all power related materials including wires, power cable, and most importantly, power supply. It has been verified in the previous section that all the components function properly. In Senior Design II, the power supply will be put to test as it will be connected to all the other components to power up KIT. One of the test methods is by programming the LEDs to be on at its brightest and see if the power supply can handle the load.

5.8 Gaming Software Tests

This section details the necessary tests that will be performed to ensure that the gaming software in functioning properly.

5.8.1 Matching Game Testing

Testing for the Matching game would consist of creating test cases to test for functionality and edge case scenarios. The initial test cases would test individual parts of the game like handling selection. Other test cases would check the ability to start and complete the game. Further test cases would measure the ability of the game to handle cases where the game might act strangely like where the selection icon would show up after the card it was selecting has been discarded. These test cases should be checked before the game is finalized. Table 33 displays some test cases.

Table 33 Test Cases

#	Test	Test Data	Expected Results	Result	Status	Note
1	Select a card and have the sprite appear correctly and return correctly	Normal game start	After selecting a card, the selected card sprite appears on screen and the user is returned to the game screen with the correct card selected			
2	Successfully discard a matched Set	Normal game start with successful match	After correctly selecting the matching card, the game should check for a game over and return you to the game screen with the next available card selected if there was no win condition			
3	Correct game over	Normal game start and competition	After the games win condition has been met, the game results screen should show with the correct time the game took			
4	Start the game and have the menu load	Normal Game Start	The game starts normally displaying the title and menu			
5	Load the game correctly	Normal Game Start	The game loads the correct board based on the difficulty			

5.8.2 Pong Variant Testing

Testing for the pong variant game would consist of test cases checking games logic. Initial test cases would check that the ball's logic performs as expected and that the players could move correctly. Other test cases would check the ability of the game to update score and to go to a win scenario. The game's edge case scenarios would be if the game could get to a loop where neither player could win, such as a ball that only bounces from side to side without moving toward either player. These cases should be tested before a final release. Table 34 displays Pong Test Cases.

Table 34 Pong Test Cases

#	Test	Test Data	Expected Results	Result	Status	Notes
1	Initial ball direction	Normal game start	The ball should start the game in the center of the screen and in a direction other than the direct left or right to prevent an inescapable loop			
2	Ball bounces correctly	The ball must bounce off a wall and/or player board	The ball should correctly reflect off a surface at a reflected angle compared to the ball's incident angle. The ball should have a different incident angle depending on where on the player board in struck			
3	Power-ups show and are activated correctly	Normal game start with power-ups acquired	No power-ups should appear at the start until the ball has struck at least one player's board. The power up should affect the last player to touch the ball. The power-up should only activate when touched by the ball			
4	Game win condition activates correctly	Complete the game	The game completes after the correct number of points has been acquired by one player. The game completion screen shows with the player scores and game time			
5	Starting the game correctly	Normal game start	The title and meu of the game show correctly			
6	Movement	Normal game mode	Both players are able to move their boards on the screen			
7	Collision	Normal game mode	The ball correctly registers a collision with the boards and the sides of the screen.			

5.8.3 Dungeon Game Testing

The testing for the dungeon game should be more extensive due to the greater scope of the game and due to the random nature of the game. The random nature of the game makes test particularly difficult since certain scenarios could occur only on certain play-throughs. More extensive testing is required to ensure the game behaves properly. The initial test cases should ensure that the creation phase of the dungeon level completes correctly. Other test cases would check the actions in the game like moving through doors or levels. Table 35 displays Dungeon Test cases.

Table 35 Dungeon Test Cases

#	Test	Test Data	Expected Results	Result	Status	Note
1	Level generation	Normal start	The level generates correctly with a number of initial points that are then connected correctly to make a level map			
2	Room generation	Normal start	Room is generated correctly within bounds and has items generated correctly			
3	Door generation	Normal start	Doors are correctly attached to rooms with the correct direction			
4	Door movement	Moving through a room door and level door	Moving through the doors transports the players to the correct room or level and the correct position on the opposite side of the door			
5	General movement	Moving though a room	Moving in a room changes the view on the game screen, that remains centered on the player			
6	Screen view	Normal start	The game screen correctly shows a map view, room view and item view that correctly shows the user’s position and items			
7	Game completion	Game win	When the user finds and moves through the dungeon exit, the game completion screen shows with the time or completion			
8	Starting the game	Normal game start	The game starts and shows the title and menu correctly			
9	Wall collision	Normal game mode	The player is unable to move past the walls of the room			

5.8.4 System Menu Testing

The testing for the Menu system needs to be the first step of integration testing for the system as a whole. If the menu system is unable to function, no other function of the K.I.T. can properly perform either. The system menu should go through several revisions based on the look and feel of the menu. Since one of the goals of a menu system is to be user friendly and intuitive to use, several renditions of the system menu are expected to be necessary. The final success or failure of this system should be based on user responses and feedback. The test sheet seen below only shows the minimum requirements for the menu and the final system could have more revisions even after these tests have been passed. Table 36 displays System Menu Test Cases.

Table 36 System menu Test Cases

#	Test	Test Data	Expected Results	Result	Status	Note
1	Reading the game files	Normal KIT start	The menu screen shows all game file names that are currently on the micro SD			
2	Showing the option screen	Normal KIT start	The user can view and use the system options menu to adjust values			
3	Main menu view	Normal KIT start	The main system menu should show a list of files that the user can scroll through and select from			
4	Performs system check	Normal KIT start/bad start	After power on, the system should perform a check on itself to ensure that all components are communicating properly			
5	Load Game	Select game from system menu	The user is able to select and properly load a game from the system menu			
6	Exit game	Game end or Exit	After a game is completed, the game can properly exit to the system menu			
7	System shutdown	Normal KIT shutdown	The user can properly shut the KIT down from the system menu			

6. Related Standards and Design Constraints

This section will go into detail on the standards and design constraints that will affect the overall project including both software and hardware related standards and constraints.

6.1 Related Standards

There are standards that need to be taken under consideration for the satisfactory of ABET. This section details the standards that are related to specific components of the project.

6.1.1 USB Standards

USB has many different standards that have evolved over the years. As time has passed, the overall design and implementation of USB has improved. As of now USB is at standard USB 3.0 which has data transfer rates of 4.8Gbit/s up from the previous standard of 480Mbps.

USB protocol states that the maximum cable length should be five meters to allow the USB module to be remote from the computer. USB has multiple different connector types (Male and Female) which tell you the right direction any two devices are supposed to be mated in. The direction is important to get correct as the remote devices is supposed to transfer data upstream to a host device while the host is supposed to transfer data downstream to the remote device. The two types of USB connectors are shown in figures 86 and 87 below.

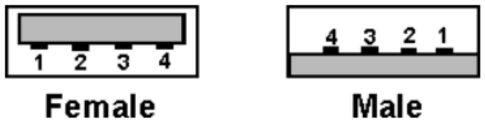


Figure 86 USB Type A

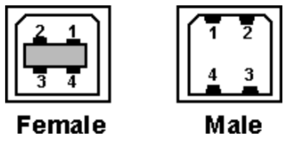


Figure 87 USB Type B

These connectors prioritize establishing the power and ground connections to be made before any signal lines are connected protecting the device from the possibility of damage to the system caused by connecting the signal lines to early. The need for small scale implementation of USB resulted in the development of Mini-USB connectors and Micro-USB connectors which both provide a compact solution to connectivity issues. These USB standards were all taken to account when deciding on the necessary connections that need to be made for the project including power transfer and data transfer [67].

6.1.2 Bluetooth Standards

All Bluetooth products fall under a specific category. These categories are Bluetooth End Product, Host Subsystem Product, Controller Subsystem Product, Profile Subsystem Product, Component Product, Development Tool and Test Equipment. A Bluetooth product is defined as any product that contains the implementation of Bluetooth technology [68].

Bluetooth has many profile standards. These profiles are protocols that help define the type of data that the Bluetooth device will send and what type of data it can receive. For this reason, for two Bluetooth modules to communicate, they must support the same Bluetooth profiles. Bluetooth implementation is generally designed for shorter distances up to 100meters. With these standards in mind, two of the same Bluetooth modules were chosen to ensure that the devices would be compatible with each other [69].

6.1.3 USART Standards

USART protocol allows for communication through a serial port connection using RS-232C protocol. It is similar to UART in protocol but allows for synchronous and asynchronous modes to be used. Differences between the asynchronous and synchronous modes include that the asynchronous mode allows for processes to work independently of each other while the synchronous mode requires both devices ends to respond to the other device before a new communication string is began. The synchronous mode will require the device to use a clock and data with the data at a fixed rate while the asynchronous mode requires only the data which does not have to be at a fixed rate.

The data will be transferred in blocks when in synchronous mode while it is transferred by individual bytes when in asynchronous mode. These things should be considered when using a device that supports the USART protocol [70].

6.1.4 SPP Standards

Serial Port Profile is a Bluetooth specific standard that aims to replace a communication interface such as the RS-232 or UART. It allows for data to be transmitted between two devices up to 100meters. The profile will allow for data to be transmitted as if receives and transmitter lines were connected between the two devices [69].

6.1.5 HID Standards

Human Interface Devices are devices that take inputs from a source and provide outputs to that source. This protocol works similarly for both USB devices and Bluetooth devices with the main difference being that USB-HID’s transmit inputs and receive the outputs through a cable while Bluetooth HID’s transmit inputs and receive the outputs through a signal. HID protocol is used primarily for devices such as mice and keyboards but is also used to create game controllers and joysticks. A HID will always have a host component and a device component. For example, a PC would be the host for a device such as a game controller [71].

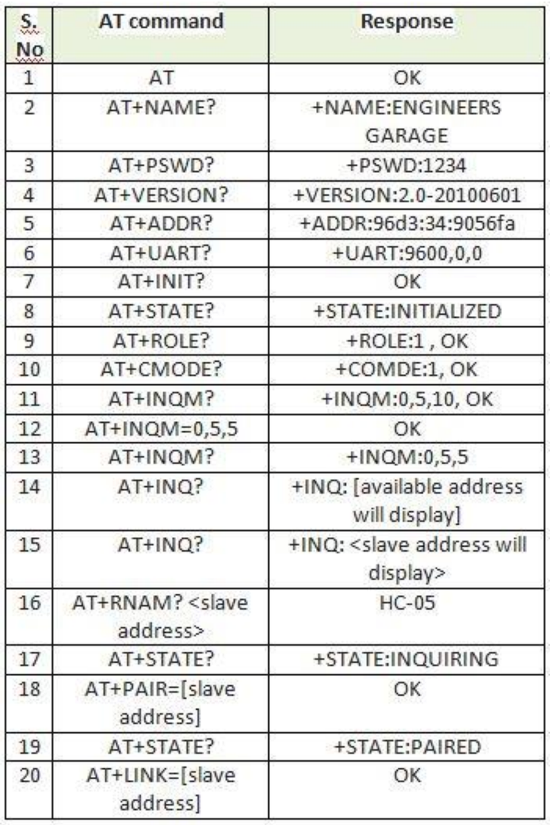
6.1.6 Arduino Nano Standards

The microcontroller uses the AVR standard. AVR microcontrollers use one eight-bit chip to operate. The chip used in the microcontroller is the ATmega328. The Arduino Nano uses the Type B Mini-USB standard discussed in a previous section. The Nano uses UART serial communication over USB in order to receive and transmit data. Because the ATmega328 is used as the Nano's chip, the Nano also supports SPI and I2C communication [72].

6.1.7 HC-05 Standards

The HC-05 uses the USART standard protocol in order to communicate with other devices. The Bluetooth module uses the SPP protocol which allows it to be compatible with many microcontrollers. The module adds full-duplex Bluetooth capabilities to any device that it is paired with [10]. The module has the ability to communicate with another HC-05 using the AT Command Mode. In this command mode, specific parameters of the module can be changed to give the module specific functionalities. These commands are displayed in the table 37 below.

Table 37 HC-05 AT Commands [102]



S. No	AT command	Response
1	AT	OK
2	AT+NAME?	+NAME:ENGINEERS GARAGE
3	AT+PSWD?	+PSWD:1234
4	AT+VERSION?	+VERSION:2.0-20100601
5	AT+ADDR?	+ADDR:96d3:34:9056fa
6	AT+UART?	+UART:9600,0,0
7	AT+INIT?	OK
8	AT+STATE?	+STATE:INITIALIZED
9	AT+ROLE?	+ROLE:1 , OK
10	AT+CMODE?	+COMDE:1, OK
11	AT+INQM?	+INQM:0,5,10, OK
12	AT+INQM=0,5,5	OK
13	AT+INQM?	+INQM:0,5,5
14	AT+INQ?	+INQ: [available address will display]
15	AT+INQ?	+INQ: <slave address will display>
16	AT+RNAM? <slave address>	HC-05
17	AT+STATE?	+STATE:INQUIRING
18	AT+PAIR=[slave address]	OK
19	AT+STATE?	+STATE:PAIRED
20	AT+LINK=[slave address]	OK

6.1.8 Power Supply Standards

In 2007, the US Department of Energy's AC/DC external power supply established an efficiency standard, requiring a strict combination of no-load power consumption and average efficiency at loads from 25% to 100% of rated load current [73]. Manufacturers are having to meet the requirement specification while keeping the low cost and producing high performing equipment. Safety standards are one of the biggest focus for power supplies including electric shock, energy hazards, fire, heat related hazards, and mechanical. The safety standards are to be considered to prevent injuring both users and manufacturers. The biggest safety goal is preventing electric shock from occurring.

According to Safety Considerations in Power Supply Design, the human body's resistance is appropriate at roughly 2000 Ohm at 110 DC volt; the human body can be impacted by the current flow [74]. Voltages above the Safety Extra Low Voltage (SELV), which is 42.4-volt peak AC or 60-volt DC coming from circuits, are considered hazardous voltage. Insulation is used as protection for power supply safety standards. The types of insulation include functional insulation, basic insulation, supplementary insulation, double insulation, and reinforced insulation; the insulating materials are either solid insulation or air insulation [74]. For a solid insulation, a 0.1 mm thickness is required if a single sheet of insulation is used.

6.1.9 Software Standards

Software standards mostly deal with code formatting and style presentation. The software standards that K.I.T. will be following mostly regard the coding method, version control, and how the code should be written. In general, the project code will follow normal code formatting conventions including Camel Case for variable names readability. Variable names must also be descriptive to their purpose instead of being singular letters. The formatting for each source file should begin with a comment block header. This header should identify the program and include a description, who the lead programmer is, and any major modification that have taken place. The initial comment block should also contain the header of all methods/functions in that source code file. All variables should also have, comments on what they are for, where they are being used, and any additional needed information, when declared. Similarly, each function should have a comment block at its declaration to identify what its use is, what it takes as parameters, and what it returns. The indentation and bracketing should follow Kernighan and Ritchie (K&R) style to be consistent across all source files [88]. The programming method being used for developing the three games for the K.I.T. is AGILE. Software version management will be handled through a central repository to hold backups for all code.

6.1.10 Screws Standards

Screw threads were first standardized back in 1864 by William Sellers of Philadelphia [75]. There are two major Unified threads series that are current in use. One is UN, and the other is Unified National Round (UNR). The specification of coarse, fine, or extra fine is specified for the UN series. On the other hand, the root radius is also needed to be specified for the UNE series [76]. The thread classifications are divided into three classes: Class 1 is ‘Loose’, the joint is frequency disassembled, Class 2 is ‘Standard’, and Class 3 is ‘Close’, for high accuracy and finer fits [76].

6.1.11 Wire and Cable Standards

The safety of wiring connections is of high importance for both the manufacturer and the users. OSHA states that a grounding terminal may never be used to reverse designated polarity or any other purposes other than grounding [77]. For a three-prong power cable, the black wire is always the hot/live wire, the white is neutral, and the green wire is the grounding wire. For safety reasons, the black and white wires should never be reversed because an internal fault could happen at any time [77]. Furthermore, it is important to not reverse the connection of the white wire and green wire since, according to OSHA, could also be dangerous [77]. Shocking may occur if the black (ungrounded) wire and the green wire are reversed even if the equipment is not operating since the wirings are wrong.

6.1.12 Raspberry Pi 2 Standards

The Raspberry Pi 2 comes with multiple USB ports, a micro USB port, an HDMI port, and a micro SD card reader [90]. The Raspberry Pi uses a quad core ARM Cortex-A7 processor as its CPU. The Pi conforms to FCC standards as a class B electronic device and conforms to ANSI standard C63.4, which provides radio noise emission standards [91]. The Raspberry Pi also comes with 27 GPIO pins. Conformation to these standards ensures that the Raspberry Pi is an excellent choice of microprocessor for the project.

6.1.13 HC-05 to HC-05 Connection Standards

The HC-05 can be used to enable a device to communicate with other Bluetooth devices. This does not restrict the HC-05 from connecting to another HC-05 module. In the case of this project, an HC-05 in the game controller must be able to communicate with the HC-05 connected to the Raspberry Pi 2. Thus, the connection standards must be discussed.

In order to connect two HC-05 Bluetooth modules, they must be configured using the AT Command Mode. This is done using an Arduino microcontroller such as the Arduino Nano. When connected to the Nano as in Figure 85, the button on the HC-05 over the enable pin is held to allow the device to enter command mode. To be able to communicate as necessary, one HC-05 must be configured to operate as a master and the other HC-05 as a slave. The AT Command to configure a HC-05 as a master is “AT+ROLE=1” while slave configuration requires “AT+ROLE=0”. Once configured, code can be written using the Arduino IDE that will allow the HC-05 master to control the outputs of the HC-05 slave based on its own inputs. In order to pair these two devices together, both modules must be configured back into data mode and re-powered. Once re-powered, the master and slave will make a connection to each other on their own with a flashing LED indicating this connection was made [93].

6.1.14 SPI standards

SPI or Serial Peripheral Interface is a communication method usually used between two different devices. SPI uses at least three lines, one for data in, one for a clock signal, and one for data out [99]. The controller that sends the clock signal is generally known as the master and any other controllers are known as slaves. There should be additional lines based on how many slaves should connect to the master. You need $\log_2 N$ lines for N slaves. SPI has the advantage of being very fast for communication but taking more lines than other communications protocols [94]. SPI is very useful to this project in particular since both Arduino and Raspberry Pi have libraries for the implementation of this protocol [95] [96]. For most communication between microcontrollers, SPI is what we should use to maintain a high speed of communication. In our project, a Raspberry Pi will run the game logic while an Arduino board will run the LED board display. To ensure a high rate of screen refresh and a responsive game, the quick speed of SPI is needed for communication between the Raspberry Pi and the other components of the K.I.T.

6.2 Design Constraints

This section details all the project constraints that affect the design of the project. It includes constraints for both the software and hardware side of the project.

6.2.1 Software Constraints

Software Constraints for the project fall under two main categories, the constraints relating to the game design and the constraints relating to the component software. Both of these types of software are important to the project and constraints need to be evaluated.

The constraints of game design in the K.I.T. system revolve around the Raspberry Pi processor. The processor will allow us to use the python programming language that has the more complicated data structures and libraries of a higher-level language. However, this means that we will not have the more lightweight C/C++ code of an Arduino. Since the Arduino relies on a C/C++ language it runs closer to the assembly level code allowing the code to be more efficient. This efficiency comes at the cost of writing object orient programming. This can make programming game logic much more difficult. Since the game programs will be written in Python, the code can take advantage of more libraries to help interact with other boards and possibly the LEDs. A disadvantage is that the code will take up more space. Another game design constraint created by the use of the Raspberry Pi is the lack of libraries directly involved with LED matrices. The Arduino does offer libraries that help control LEDs, control matrices, and libraries to help with button debounce that could be useful. However, the ability to program object-oriented code will allow much greater complexity in the code produced.

Another game design constraint will be the LED matrix display itself. The LED matrix will prevent high resolution images from being used in any of the games produced for the K.I.T. This is not necessarily a bad thing since it also allows the display to be simpler to program. The main issue will be ensuring the games have enough fidelity to ensure the user knows what is going on. This could force some games to have constraints like leaving a line of LEDs blank between different sections of the screen to ensure the user knows they are separate. This could also limit how many pixels a sprit can move across the screen at a time. To counter this, the use of RGB LEDs was chosen to allow the games more options in design.

Another possible constraint is the display of text on the LED matrix display. Because of the limits of the LED display the text written to the display would have to be at least five by five pixels large to ensure fidelity. To ensure this design a simple representation of numbers 0-9 and letters A-Z was created to ensure that text could be quickly displayed on the screen. However, the number of LEDs meant that only twenty-four numbers or letters could be displayed on the screen at one time. The use of scrolling text is needed to ensure that the full length of any message can be displayed. This could be a constraint to some game types or game design forcing programmers to use less text to save screen space.

The constraints associated with the component software involve learning how to program the individual component boards. For any Arduino boards, the C code will allow the software to be lightweight and simple, but also constrain its complexity. However, some boards might require a more in-depth learning process to properly code for. Depending on the board it might be necessary to program in assembly which can be difficult. Another constraint is ensuring that the component's software can communicate to one another. For components that communicate through Bluetooth or Wi-Fi, there is a defined protocol. For other components, the software might constrain the method or type of communication from device to device.

6.2.2 Environmental Constraints

The importance of environmental constraints should be taken under consideration in any project. The following sections discuss how the selected components will affect the environments.

6.2.2.1 LCD Screen Disposal

Although Liquid Crystal Display (LCD) has become a preferred display device as it consumes low power and does not take up a lot of spaces, it may have a negative impact on the environment once the LCD's lifetime is ended if not disposed properly. This is due to the presence of mercury and liquid crystals which are not easy to biodegrade [85]. The mercury is found in the backlight unit of LCD. It is toxic to soil and water in warmer weather condition [85]. There are also hazardous gases found within the production of LCD screens. These gases include, but not limited to, silane, phosphine, ammonia, and hydrogen [85]. The exposures to these gases may cause cardiovascular and respiratory issues [85]. The best way to dispose of an LCD screen is through recycling.

6.2.2.2 Game Controller Disposal

The game controller shell and button covers are made of PLA or TPE filament which are plastic based substances. Thus, their environmental impact should be discussed in case there is a decision to dispose of the shell components. TPE filament is the most environmentally friendly filament type as it is processed from raw, renewable materials. The best way to dispose of this material is to recycle it as it has short cycle times which reduces waste [87]. PLA filament can be recycled as well but must be recycled separately as it is plant based and must be sent to a composting facility. Although recyclable, PLA biodegrades very slowly and may take 100 to 1000 years to decompose in a landfill. Thus, it must be recycled in a controlled composting environment to ensure that it biodegrades within three months [86].

6.2.2.4 LED Disposal

LED bulbs contain heavy metals such as lead and copper. Currently there is no regulation toward the disposal of LEDs. They are not classified as hazardous waste and so can be disposed of through residential trash collection. [105]

6.2.3 Health and Safety Constraints

The safety of manufacturers and users should always be taken as first priority when producing a new product. The following sections discuss the various health and safety concerns that are related to the product.

6.2.3.1 Power Cord Safety

Ones of the highest cause of electrical hazards are shocking and electrocution. It is very importance that the device is properly grounded especially the three-prong power cord. With the utilization of an ON/OFF switch of K.I.T., care must be taken in securing the connections between the two live wire terminals and that the ground and neutral wires are secured all the way through. All copper wires should not be exposed and should be covered using insulator materials. If it is discovered that there is a present of the copper wires, electrical tape can be used to cover the wires and insulate them.

6.2.3.2 Game Controller Safety

As the game controller is comprised of mainly electronic components powered by a battery, precautions must be taken when handling the controller. At no point in time should the game controller be exposed to any liquid of any sort. Exposure to liquids provides the possibility of electrocution and damage to all of the electrical components in the game controller. When the game controller is connected via USB to the K.I.T, it is imperative that the controller is not exposed to liquid as well as there are even more electrical components involved that can cause damaging shocks to the system and user.

6.2.3.3 LED Board Safety

The LED board is the main graphical display for the K.I.T. It is expected that the user looks at the LED board for extended periods of time while they are gaming. Due to the brightness of these LEDs, a diffuser is going to be used on the LED board to reduce the bright intensity of the LEDs while the K.I.T. is being used. The diffuser should not be removed during normal use of the K.I.T. Due to the brightness of the LEDs, they should not be viewed directly by the user. This is why both the diffuser and a LED brightness knob are features of the K.I.T. IEEE standards on LED safety have recommendations on dimming and modulating brightness to ensure user safety [100]. This information will be taken into account when determining the default brightness for the K.I.T. system.

There are also International standards for LED general safety standards. These standards deal with things like heat management, overpower conditions, LED durability, and moisture resistance [101]. These concerns should be taken into consideration when designing the casing of the K.I.T. to ensure that the LEDs are properly maintained and can run for extended periods of time. The heat generation of the LEDs could affect other components in the K.I.T. and possibly affect performance. Endurance testing might need to be considered to ensure that the system can run consistently under stress. The LEDs should also not be exposed to moisture to maintain proper use. With these safety standards in mind we can ensure the user knows how to properly run our system and how to properly design it.

The LED standards are focused on eye safety. It is important that the user cannot be harmed by bright lights or made ill from the display. LEDs are capable of producing powerful light intensity and caution must be exhibited to ensure photobiological safety. At a high intensity any light source has the potential to be harmful to both the skin and the eyes through ultraviolet radiation. Testing indicates that blue and royal blue LED components in the 450-485 wavelength range pose significant eye safety risks. [104] The product safety standard applicable to the LED is the IEC 61010. This standard indicates that only Class 1 or Class 2 laser products should be used for demonstration, display, or entertainment in unsupervised areas. This ensures that spectators are prevented from exposure to levels exceeding the applicable MPE. If the demonstration product is used for educational purposes it must comply with all applicable requirements detailed in the IEC 61010. The equations that define the different class laser are extensive. To summarize the information the maximum emitted power for Class 1 is 0.56 mW. This class is eye-safe under all operating conditions. Along with the KIT, it is imperative that there is practical labeling to prevent injury. The accepted standard is an explanatory label stating "CLASS 1 LASER PRODUCT" for class 1 and "LASER RADIATION, DO NOT STARE INTO BEAM, CLASS 2 LASER PRODUCT" for class 2 products. Class 2 is safe for accidental viewing under all operating conditions. However, it may not be safe for a person who deliberately stares into the laser beam for longer than 0.25 s, by overcoming their natural aversion response to the very bright light [103]. The KIT avoids harming users by utilizing a diffuser portion to reduce radiation by reflecting light and reducing concentration.

6.2.4 Game Controller Constraints

The physical game controllers are designed such that there are eight push buttons. Four of these buttons were designed for directional movement on the K.I.T while the other four were designed for selection. The number of buttons that can be designed on the game controller is directly affected by what microcontroller is used to drive the game controller. The Arduino Nano was chosen as the main microcontroller for the game controller. It has twenty-two input/output pins that can be used. Taking into account that certain pins must be used to connect the Arduino Nano to the HC-05 Bluetooth module, not all of these pins can be used to connect push buttons. Thus, if the project required more push buttons for game interactions, it may be impossible depending on how many extra push buttons are needed. If there were not enough input/output pins to support the needed extra push buttons, a different microcontroller that has the extra pins would have to be used.

The game controllers can be connected to the K.I.T in multiple ways. This includes a USB connection and Bluetooth connection. With these connection types comes constraints on the distance you can be from the K.I.T while still being able to play it. When using a USB connection, the game controller can only be taken about 4-5 feet away from the K.I.T as that is the length of the USB cable. When using a Bluetooth connection, the user can stand up to one-hundred meters away from the K.I.T while still being able to play.

6.2.5 LED Display Constraints

The LED chosen for display is the WS2812B. This is an RGB LED component. These constraints the color capabilities of the display as it is not capable of producing true white among other colors. Reducing the color capabilities reduces the ability of the game designer to implement shading in graphic designs. Because the simple display type this constrain should not significantly impact the designer as it is not designed for any highly detailed art or animation in game play.

6.2.6 Audio Constraints

The components chosen for the KIT do not allow for built in speakers. The power consumption would be too great if an amplification component was utilized a built-in speaker component. Also, the current architecture only allows for the mixing of 3 different audio signals. This constrains the designer in how many audio feedback signals can be implemented at a single time. For example, If the designer had a long audio signal following a user input such as the sound of a race car driving around the track, the background audio of the gameplay, the audio of competitor vehicles driving by, and an addition sound would not be possible. The gameboard is only capable of mixing a select amount of audio signals. Designing a game in which 4 different audio signals are mixed simultaneously is not possible.

7. Administrative Content

This section outlines the team's finance and milestones that will allow the team to meet the deadlines of Senior Design I and II.

7.1 Project Budget and Financing

The table 38 below contains a list of the products that have been purchased ever since the beginning of Senior Design I. The total budget for the entire budget was set to the limit of $800. Currently, the group has spent the total of less than $340. There are still minor components that will be bought next semester during Senior Design II when the project is being put together.

Table 38 Financial Budget

Product	Quantity	Price
14-16 AWG Terminals Ring	1	$7.99
16 AWG 3 Prong Power Cord	1	$18.09
Adafruit Perma-Proto Half-sized Breadboard PCB - 3 Pack	1	$16.07
Audio board	1	$14.25
Bus Transceivers 74HC	3	$1.77
DSD TECH HC-05 Bluetooth	2	$17.98
Electrical Wire 16 AWG	1	$8.80
ELEGOO 3PCS 400 tie-points breadboard	1	$8.21
ELEGOO for Arduino Nano V3.0	1	$13.86
LED 60/m WS2812B	3	$64.93
Leviton 5410 Appliance Switch	1	$4.41
Mean Well RSP-320-5	1	$48.10
Modular Connectors / Ethernet Connectors RJ45 Connector	6	$15.30
Multilayer Ceramic Capacitors MLCC	5	$1.15
Raspberry Pi 2	1	$40.00
Tactile Push Button	1	$9.99
Teensy 3.2 USB Development Board	2	$39.60
Thick Film Resistors	30	$1.98
Total		**$332.48**

7.2 Project Milestones

The following tables detail the goals set by the senior design team in order to meet all deadlines required for both Senior Design I and II. Table 39 shows the milestones set out for Senior Design I.

Table 39 Senior Design 1 Milestones Schedule

Senior Design 1	
Week	**Description**
1-2	Brainstorm ideas
3-4	10 pages Divide and Conquer document
5-10	Research
11-15	60 pages turn in
16-17	Finalize part list/Order parts
18-20	100 pages turn in
21-24	120-page final turn in

So far, the set milestones for Senior Design I has been continuously met throughout the entire semester. More tests have been executed more than needed, this should put the team ahead of the schedule for Senior Design II for the next semester. The Table 40 below details the milestones for Senior Design II.

Table 40 Senior Design 2 Milestone Schedule

Senior Design 2	
Week	**Description**
1-5	Assemble prototype (MAP)
6-8	Refine design of prototype
9-10	Design review 1
11-14	Testing prototype
15-16	Design review 2
17-20	Testing prototype
21-23	Finalize design
24	Demonstration

8. Summary and Conclusion

The Knights Interaction Timebox is constructed with the classic game player in mind. From the retro controllers, simple gameplay, and retro game types the project meets all marketing expectations. The simple LED display used to achieve full color yet simplify game display for developers has passed initial tests. The control components utilizing the Arduino and Bluetooth module combination mix the classic game controller feel with newer wireless technology. The Raspberry Pi is a staple in retro gaming emulation and allows for developers to get innovative in simple game design.

The LED game board, K.I.T., will be manufactured with users' wants and needs in mind. The release of the K.I.T. is planned to have three games written within the software that are stored in a micro SD card. The software will be transferable via a micro SD card slot that will be located on the side of K.I.T. device. The users will have the ability to interact with the board using handheld controllers that come with 4 directional buttons and 4 selection buttons, in a similar way as the traditional game controllers. The controllers can be connected to the board via a USB port that is connected to the Raspberry Pi 2. They can also be connected via Bluetooth with a connection between HC-05 Bluetooth modules. The HC-05 Bluetooth module connected to the Raspberry Pi 2 will allow the user to use an Arduino app as a game controller as well.

Some of the other features of K.I.T. include resolutions of 24 by 32 RGB LEDs, giving the total of 768 LEDs within the board, the LCD will be installed on the side of the board to display the Bluetooth connection status along with the internal temperature of K.I.T. The brightness of LEDs will be adjustable allowing users to set their screen brightness to their satisfaction. Without having to physically unplug the power cord from an AC outlet, K.I.T. will contain an ON/OFF switch that is close to the actual device.

The user will have the option of playing three distinct games on the K.I.T. The Matching game, Pong and Dungeon game will each give the user a different gaming experience with Pong allowing for the sole multiplayer experience. Each of these games will be accessible from the K.I.T. startup menu and from an in-game menu. Because of the micro SD card used to store these games, more games can be added later if desired.

The project is on pace to be completed before the deadline with testing and prototyping already underway. All major components have been received and ran through basic verification testing. Game logic has been developed and component interfaces have been outlined. Currently the project is operating under budget and set to meet all requirements and expectations.

9. Appendices

The following sections include the sources that are used for research related materials and the permissions from manufacturer to use the schematics that are included throughout the document. Some of the manufacturer have not responded to the emails that were sent but the screenshots of the emails are included in the section below to provide proof that the team has reached out to the manufacturer to try and obtain permission for usage of the schematics.

9.1 Bibliography

[1] https://all3dp.com/2/tpe-filament-explained-and-compared/
[2] https://www.allthat3d.com/pla-vs-abs/
[3] https://cdn-learn.adafruit.com/downloads/pdf/introducing-bluefruit-ez-key-diy-bluetooth-hid-keyboard.pdf?timestamp=1542251963
[4] https://www.adafruit.com/product/1535
[5] https://cdn-learn.adafruit.com/downloads/pdf/introducing-the-adafruit-bluefruit-le-uart-friend.pdf?timestamp=1542253659
[6] https://learn.adafruit.com/introducing-the-adafruit-bluefruit-le-uart-friend/introduction
[7] https://store.arduino.cc/usa/arduino-uno-rev3
[8] https://i.chillrain.com/index.php/arduino-uno-and-nano-pinout-diagram/stampa/
[9] https://store.arduino.cc/usa/arduino-nano
[10] https://components101.com/wireless/hc-05-bluetooth-module
[11] https://www.sparkfun.com/products/12574
[12] https://store.arduino.cc/usa/arduino-micro
[13] http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0464f/index.html
[14] https://www.raspberrypi.org/documentation/usage/gpio/README.md
[15] https://store.arduino.cc/usa/arduino-mega-2560-rev3
[16] https://www.waveformlighting.com/home-residential/how-to-choose-a-power-supply-for-your-led-strip-project
[17] https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf
[18] http://www.ledlightsworld.com/dc-12v-dimmable-smd5050300-flexible-led-strips-60-leds-per-meter-10mm-width-900lm-per-meter-p-92.html
[19] https://www.raspberrypi.org/documentation/faqs/#pi-power
[20] https://www.adafruit.com/product/2756
[21] http://www.electronicaestudio.com/docs/istd016A.pdf
[22] http://wiki.sunfounder.cc/index.php?title=LCD2004_Module
[23] https://www.beta-estore.com/download/rk/RK-10290_410.pdf
[24] https://gdstime.manufacturer.globalsources.com/si/6008847878553/pdtl/Cooling-fan/1156874012/Brushless-Axial-Cooling-Fan.htm
[25] http://www.meanwellusa.com/webapp/product/search.aspx?prod=RSP-320
[26] http://www.meanwellusa.com/webapp/product/search.aspx?prod=HRP-300
[27] http://www.meanwellusa.com/webapp/product/search.aspx?prod=HRPG-300
[28] https://www.stayonline.com/product-resources/reference-circuit-ampacity.asp

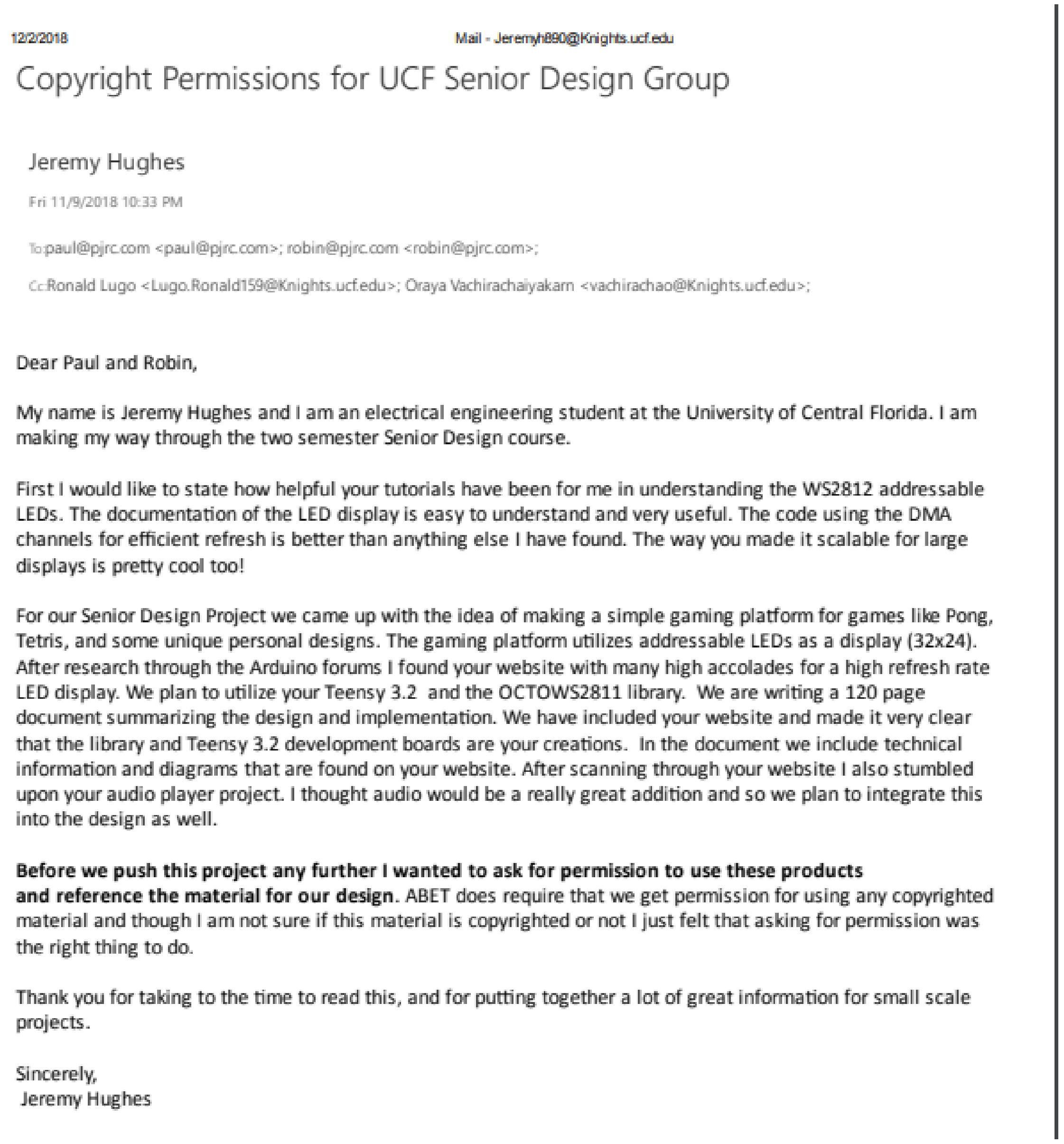
[29] https://www.allaboutcircuits.com/textbook/reference/chpt-2/wiring-color-codes/
[30] https://www.anixter.com/en_us/resources/literature/wire-wisdom/cable-jackets-types-101.html
[31] https://www.edn.com/electronics-blogs/power-supply-notes/4415668/What-size-and-type-of-output-wires-should-I-use-
[32] https://learn.adafruit.com/adafruit-audio-fx-sound-board/overview
[33] https://cdn-shop.adafruit.com/datasheets/vs1000.pdf
[34] https://georgehill-timber.co.uk/blog/mdf-vs-plywood-for-projects/
[35] https://www.bobvila.com/articles/mdf-vs-plywood/
[36] https://www.engineering.com/Library/ArticlesPage/tabid/85/ArticleID/152/categoryId/11/Thermal-Conductivity.aspx
[37] https://www.decospan.com/media/files/en-decospan-decopanel-2.pdf
[38] http://www.woodworkbasics.com/mdf.html
[39] https://www.foamboards.com.au/What-is-Foam-Board
[40] http://www.greenspec.co.uk/building-design/insulation-materials-thermal-properties/
[41] https://www.creativemechanisms.com/blog/injection-mold-3d-print-cnc-acrylic-plastic-pmma
[42] http://www.madehow.com/Volume-2/Acrylic-Plastic.html
[43] https://www.creativemechanisms.com/blog/everything-you-need-to-know-about-polycarbonate-pc
[44] https://www.curbellplastics.com/Research-Solutions/Materials/Polycarbonate-Film
[45] https://thermtest.com/applications/insulation-vs-thickness-thermal-conductivity-hfm
[46] https://sciencing.com/info-8789172-thermal-properties-cardboard.html
[47] https://www.azom.com/article.aspx?ArticleID=1446
[48] https://sciencing.com/aluminum-vs-steel-conductivity-5997828.html
[49] https://www.fastenermart.com/files/wood-screw-characteristics.html
[50] http://www.craftsmanspace.com/knowledge/dowelling-dowel-joint.html
[51] https://www.mybluprint.com/article/5-types-of-wood-glue-what-to-know-how-to-use-them
[52] https://www.leviton.com/en/products/5410
[53] https://cpldcpu.wordpress.com/2014/01/14/light_ws2812-library-v2-0-part-i-understanding-the-ws2812/#comments
[54] https://www.pjrc.com/teensy/td_libs_OctoWS2811.html#tech
[55] https://www.pjrc.com/store/octo28_adaptor.html
[56] https://www.pjrc.com/teensy/td_libs_OctoWS2811.html
[57] https://www.pjrc.com/teensy/teensy31.html
[58] https://assets.nexperia.com/documents/data-sheet/74HC_HCT245.pdf
[59] https://www.arctic.ac/us_en/f12-silent.html
[60] http://www.gdstime.com/product/?99_481.html
[61] https://www.raspberrypi.org/
[62] https://www.arduino.cc/reference/en/ - Arduino reference library
[63] https://www.pygame.org/docs/
[64] http://www.ti.com/lit/ds/symlink/tlc5940.pdf - TI PWM LED driver data sheet
[65] https://www.adafruit.com/product/938 - mini LCD display from adafruit
[66] https://www.jsumo.com/hc-05-bluetooth-module-serial-transceiver-module

[67] https://www.electronics-notes.com/articles/connectivity/usb-universal-serial-bus/standards.php
[68] https://www.bluetooth.com/specifications/bluetooth-core-specification
[69] https://learn.sparkfun.com/tutorials/bluetooth-basics/bluetooth-profiles
[70] https://whatis.techtarget.com/definition/USART-Universal-Synchronous-Asynchronous-Receiver-Transmitter
[71] https://en.wikipedia.org/wiki/Human_interface_device
[72] https://www.mouser.com/pdfdocs/Gravitech_Arduino_Nano3_0.pdf
[73] https://www.edn.com/design/power-management/4436816/AC-DC-power-supply-performance-and-international-efficiency-standards
[74] https://www.ieee.li/pdf/essay/safety_considerations_in_power_supply_design.pdf
[75] https://www.asme.org/about-asme/who-we-are/engineering-history/landmarks/234-the-united-states-standard-screw-threads
[76] http://portal.ku.edu.tr/~cbasdogan/Courses/MDesign/course_notes/fasteners.pdf
[77] https://www.osha.gov/dte/library/electrical/electrical.pdf
[78] www.digitaltrends.com
[79] www.businessinsider.com
[80] www.livescience.com
[81] www.mogi-translations.com
[82] https://cdn-shop.adafruit.com/product-files/181/p181.pdf
[83] https://www.adafruit.com/product/181
[84] https://www.adafruit.com/product/1115
[85] http://www.ijestr.org/IJESTR_Vol.%201,%20No.%207,%20July%202013/Liquid%20Crystal%20Display.pdf
[86] https://www.thoughtco.com/pros-cons-corn-based-plastic-pla-1203953
[87] http://advantage-environment.com/buildings/thermoplastic-elastomers-flexible-and-recyclable/
[88] https://en.wikipedia.org/wiki/Indentation_style#K.26R
[89] https://www.arduino.cc/en/Main/FAQ#toc11
[90] https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md - raspberry pi hardware reference
[91] https://www.raspberrypi.org/documentation/hardware/raspberrypi/compliance/rpi_DOC_2b_FCC.pdf
[92] https://fossbytes.com/arduino-remote-control-apps-android/
[93] https://howtomechatronics.com/tutorials/arduino/how-to-configure-pair-two-hc-05-bluetooth-module-master-slave-commands/
[94] https://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/
[95] https://www.arduino.cc/en/Reference/SPI
[96] https://www.raspberrypi.org/documentation/hardware/raspberrypi/spi/README.md
[97] https://opensource.com/life/16/3/how-configure-raspberry-pi-microcontroller
[98] https://theomandel.com/resources/golden-rules-of-user-interface-design/
[99] https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all
[100] https://webstore.ansi.org/Standards/IEEE/IEEE17892015
[101] https://webstore.ansi.org/Standards/IEC/IEC62031Ed2018

[102] https://www.engineersgarage.com/contribution/pairing-two-hc-05-bluetooth-modules-and-share-data-between-them
[103] https://www.lasermet.com/resources/classification_overview.php
[104] https://www.cree.com/led-components/media/documents/XLamp_EyeSafety
[105] http://www.lessismore.org/materials/278-led-lights
[106] https://www.pjrc.com/store/teensy3_audio.html
[107] https://www.digitaltrends.com/gaming/pc-market-grew-in-2016-led-by-mobile-and-pc-gaming/
[108] https://www.businessinsider.com/nintendo-nes-classic-edition-sales-vs-ps4-xbox-one-2018-8
[109] http://www.mogi-translations.com/top-10-best-selling-video-games-of-all-time/
[110] https://www.livescience.com/56481-strange-history-of-tetris.html
[111] http://www.eecs.ucf.edu/seniordesign/sp2015su2015/g14/
[112] https://www.arduino.cc/reference/en/ - Arduino reference library
[113] http://wiring.org.co/reference/libraries/Matrix/index.html - Arduino library to help code LED matrices
[114] https://learn.adafruit.com/introducing-the-adafruit-bluefruit-le-uart-friend/pinouts
[115] https://components101.com/wireless/hc-05-bluetooth-module
[116] https://www.parallax.com/sites/default/files/downloads/P8X32A-Propeller-Datasheet-v1.4.0_0.pdf
[117] http://www.world-semi.com/DownLoadFile/108
[118] https://www.witop-tech.com/product-item/newest-cs8208-digital-rgb-12v-individually-addressable-rgb-led-strip/
[119] http://www.world-semi.com/DownLoadFile/136
[120] http://cache.freescale.com/files/32bit/doc/data_sheet/K20P64M72SF1.pdf

9.2 Permissions

Some of the design schematics are being referenced from a site. We have emailed the people who designed those schematics and codes to ask for permission to use their materials as reference. Figure 88 below is proof that we have seek out to ask for permission.



12/2/2018 Mail - Jeremyh890@Knights.ucf.edu

Copyright Permissions for UCF Senior Design Group

Jeremy Hughes

Fri 11/9/2018 10:33 PM

To:paul@pjrc.com <paul@pjrc.com>; robin@pjrc.com <robin@pjrc.com>;

Cc:Ronald Lugo <Lugo.Ronald159@Knights.ucf.edu>; Oraya Vachirachaiyakarn <vachirachao@Knights.ucf.edu>;

Dear Paul and Robin,

My name is Jeremy Hughes and I am an electrical engineering student at the University of Central Florida. I am making my way through the two semester Senior Design course.

First I would like to state how helpful your tutorials have been for me in understanding the WS2812 addressable LEDs. The documentation of the LED display is easy to understand and very useful. The code using the DMA channels for efficient refresh is better than anything else I have found. The way you made it scalable for large displays is pretty cool too!

For our Senior Design Project we came up with the idea of making a simple gaming platform for games like Pong, Tetris, and some unique personal designs. The gaming platform utilizes addressable LEDs as a display (32x24). After research through the Arduino forums I found your website with many high accolades for a high refresh rate LED display. We plan to utilize your Teensy 3.2 and the OCTOWS2811 library. We are writing a 120 page document summarizing the design and implementation. We have included your website and made it very clear that the library and Teensy 3.2 development boards are your creations. In the document we include technical information and diagrams that are found on your website. After scanning through your website I also stumbled upon your audio player project. I thought audio would be a really great addition and so we plan to integrate this into the design as well.

Before we push this project any further I wanted to ask for permission to use these products and reference the material for our design. ABET does require that we get permission for using any copyrighted material and though I am not sure if this material is copyrighted or not I just felt that asking for permission was the right thing to do.

Thank you for taking to the time to read this, and for putting together a lot of great information for small scale projects.

Sincerely,
Jeremy Hughes

Figure 88 Permission for Reference Materials