UCF

COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

MULTI-PURPOSE DEVICE CONTROLLER

Group 36 - Authors:

Johnny Blanc	**Computer Engineering**
Jason Jean-Louis	**Computer Engineering**
Mohammed Talib Hamed Rashid Al Fahdi	**Electrical Engineering**
Kenny Dao	**Computer Engineering**

Table of Contents

Figures

Tables

1.0 Executive Summary

In the realm of controller design, the persistent issue of "stick drift" has plagued users, affecting precision and usability. To address this, our project focuses on creating an innovative Bluetooth universal controller utilizing hall effect joysticks powered by an ESP32 microcontroller.

The core challenge we tackle is the reliability and longevity of controller inputs. By integrating hall effect joysticks, which employ magnetic fields for precise positioning, we aim to eliminate stick drift entirely. This technology ensures consistent and accurate user input, enhancing the overall gaming or device-controlling experience.

Our controller design leverages the capabilities of the ESP32 microcontroller, offering a seamless Bluetooth connection for versatile compatibility across various platforms. The implementation of Bluetooth technology ensures wireless connectivity, providing users with freedom of movement and flexibility in usage scenarios.

The project's key features include a robust and ergonomic design that prioritizes user comfort and durability. Furthermore, the integration of hall effect sensors guarantees extended longevity and accuracy in input mechanisms, setting a new standard for controller reliability.

Our development process involves meticulous prototyping and rigorous testing to ensure optimal performance and compatibility across a wide range of devices and applications. User feedback and iterative improvements are integral parts of our methodology, aiming to deliver a superior, user-friendly controller.

The project's budgetary allocation is structured to ensure efficient resource utilization while maintaining high-quality standards. Milestones are established to track progress, ensuring timely completion while adhering to budget constraints.

Our project's documentation spans eleven chapters, each written to cover crucial aspects of our development journey. It begins with an executive summary encapsulating our objectives, motivation, and goals, followed by an expansive project background and motivation outlining our goals, existing work, and engineering specifications. Chapter three delves into research, comparing technologies and parts, and four explore standards and design constraints in detail. A comparison of ChatGPT with similar platforms follows, emphasizing learning experiences. Hardware and software design chapters elaborate on system architecture, while system fabrication, testing, and administrative content outline the practical implementation, testing methodologies, budget, and work distribution. Finally, a concise conclusion summarizes key outcomes and future considerations. Each chapter represents a vital segment, collectively providing a holistic view of our project's inception, evolution, and plans for realization.

In conclusion, our project represents an advancement in controller technology, addressing the prevalent issue of stick drift through innovative hall effect joysticks and the versatility of Bluetooth connectivity. With a focus on precision, reliability, and user-centric design, our

Bluetooth universal controller aims to redefine the standard for user interface devices across various applications and industries.

2.0 Project Background and Motivation

2.1 Introduction

This paper embarks on a comprehensive exploration of our senior design project – the Multi-Purpose Device Controller (MPDC). We delve into the intricate research, functionality, and procedural steps involved in designing a controller that closely resembles a gaming controller but possesses the capability to seamlessly connect to a wide range of devices, including gaming consoles, laptops, and more. Our goal is not only to simplify the user experience but also to address pressing factors that make the MPDC a compelling solution in today's technological landscape.

2.2 Product Comparison

Given what is on the market, the idea of a cross-platform controller is not a novel concept, there is one controller on the market already called The ALL Controller, which was made to be connected to any PC, Mac, Linux, Xbox 360, Xbox One, PlayStation 3, PlayStation 4, Android, iOS, and Nintendo Switch. Making this a very versatile controller for all scenarios. Connecting to other consoles through a USB adapter or through Bluetooth, retailing at $65 before this seemed to be a solid contender on the market, however, this controller has not been updated to be compatible with newer-gen consoles; production has also stopped for these specific controllers all together, given what it was capable of, that is a very unfortunate reality.
Another controller that hit the market with multi-platform functionality was The 8Bitdo Pro 2, with wireless compatibility with Android devices, iPhones, Macs, PCs, and the Nintendo Switch. While it is an excellent controller for mobile, PC, and Switch gamers for an affordable price, retailing at $50; on the opposite end, it offers slightly limited programmable macros and lacks a couple of Switch-specific features.

Then finally there is the scuf gaming controller, which is limited in its cross-platform potential, being only usable with one of the major consoles (Xbox or PlayStation), and PC. But what they lack in compatibility, they make up for in customization, letting you accessorize everything from the button faces to the skin of the controller itself. Those controllers also offer button remapping on the fly using a magnet. While the scuf controllers offer a lot, the price is not very consumer-friendly, with the prices varying from $150- $250 depending on how it is customized. Also, the longevity of each controller is not great, the controller would only last for a year before it starts to develop issues such as stick drift or a faster-draining battery, depending on its usage.

Our goal with this project is to combine the versatility of the ALL controller, the on-the-go remapping capabilities of the Scuf controller, and the affordability of the 8Bitdo Pro 2. By amalgamating these key features, we aim to craft a controller that transcends the limitations of current offerings in the market. Our vision is to create a device that seamlessly adapts to any gaming platform, ensuring compatibility with a vast array of games and applications. Furthermore, we're committed to making this innovation accessible to a wide audience without imposing a hefty financial burden. Through meticulous design and engineering, we aspire to provide a controller that not only elevates the gaming experience but also stands as a testament to our dedication to innovation and consumer satisfaction. With this endeavor, we're poised to revolutionize the gaming accessory landscape, setting a new standard of excellence and accessibility for gamers worldwide.

2.3 Statement of Motivation

In an era marked by rapid technological advancement and an ever-expanding array of interconnected devices, the demand for a versatile and user-friendly control solution, exemplified by the Multi-Purpose Device Controller (MPDC), has never been more apparent. Our world is teeming with gadgets, IoT devices, and appliances, each featuring its distinct and often convoluted control interface. This proliferation of control mechanisms can be overwhelming and, at times, exhausting for users seeking a straightforward and efficient means to manage their devices. It is this pressing need for simplicity and enhanced user experience, as epitomized by the MPDC, that serves as the driving force behind our senior design project.

In today's economic climate, where efficiency and cost-effectiveness are paramount, the development of the MPDC holds immense promise. By consolidating control functions for a diverse range of devices into a single, user-friendly interface, our project aims to offer consumers a cost-effective and accessible solution that aligns with their evolving needs. This, in turn, has the potential to boost economic efficiency by reducing the need for multiple, often expensive, specialized controllers.

Furthermore, the environmental impact of our endeavor cannot be understated. In a world increasingly conscious of its ecological footprint, the MPDC's design philosophy encourages the use of a single, multifunctional device, contributing to the reduction of electronic waste and the overall conservation of resources. This environmentally conscious approach aligns with the global imperative to minimize the environmental impact of consumer electronics.

In summary, our senior design project, centered around the development of the Multi-Purpose Device Controller (MPDC), seeks to address the contemporary challenges posed by an interconnected world, a shifting economy, and growing environmental concerns. We are motivated by the opportunity to simplify and enhance the user experience, promote economic efficiency, and contribute to a more sustainable future through the creation of a versatile controller capable of seamlessly connecting to a variety of devices.

2.3.1 Goals

Our main goal for this project is to develop a multipurpose controller that offers universal compatibility, customization options, and exceptional durability. This versatile controller is envisioned to connect and interact seamlessly with a variety of devices, mainly gaming consoles and computers. Recognizing the diversity of user preferences and needs, we aim to provide robust customization options, allowing users to tailor the controller's layout and functionality to suit their specific requirements. Additionally, our design will prioritize durability and ergonomic comfort, ensuring that users can rely on the controller for both short interactions and extended gaming or entertainment sessions.

2.3.2 Objectives

We plan to achieve universal compatibility for a gaming controller, to execute this goal we begin the process by identifying target platforms such as PCs, gaming consoles (e.g., Xbox, PlayStation), mobile devices (iOS and Android), and potentially other systems like VR platforms. Then it is crucial to understand compatible standard communication protocols like USB, Bluetooth, and HID, which are prioritized to ensure broad support. Multiple communication interfaces will be implemented within the controller to cater to diverse platforms, utilizing USB for PCs and Bluetooth for mobile devices, for example; Platform-specific device drivers will be developed for each target platform, potentially through collaboration with platform manufacturers. These drivers will be certified and digitally signed, particularly for platforms with strict requirements like Windows, to enhance compatibility and security. Additionally, mechanisms for automatic driver installation and regular updates will be established. An Abstract layer that will be implemented in the MPDC is a Device Driver Translation Module (DDTM) within the controller's firmware that can differentiate between platforms to allow integration with platform-specific APIs and SDKs. Rigorous testing, including diverse platform and device testing, along with comprehensive documentation and support will also be implemented to further enhance compatibility. Continuous monitoring of platform changes and timely firmware and driver updates will be done to address evolving compatibility changes.

2.3.3 Stretch Goals

We have ambitious stretch goals for our gaming controller project, envisioning multiple prototypes with unique features to cater to diverse gaming preferences. One of our prototypes will incorporate a touchpad that not only functions as an additional button but also offers touch capabilities, enhancing gameplay versatility. Another variant will boast back panel buttons, providing an intuitive way to access controls without the need to remove your thumb from the joystick, all while offering vibration feedback for an immersive experience. Our most innovative concept includes a small display screen, replacing the touchpad, and enabling seamless switching between different button configurations on the fly. In addition, we aim to develop a dedicated mobile app that empowers users to save and effortlessly switch between personalized button configurations, offering a user-friendly and flexible customization experience. These stretch goals showcase our commitment to creating a gaming controller that truly adapts to the unique preferences of gamers. Lastly, we also want to improve the power efficiency of the MPDC. We estimate that the power dissipation for the MPDC will be around 2.5 W. However, with an added component called Bluetooth low energy, we are hoping we can reduce power efficiency by at least 10%.

2.4 List of Specifications for The Project as a Whole

Type	Component
User Input	● 2 × analog sticks (anti stick-drift) ● 2 × analog triggers ● 2 × shoulder buttons ● 4 × action buttons
Power	● USB-C Charger ● Power bank: Voltage: 5v Current: 5000mAh ● Voltage regulator: 5v to 3v3 ● LEDs: Voltage input: 5v
Dimensions	● (150-175 mm) × (100-120 mm) × (50-70 mm)
Weight	● 250-300g
BLE(Bluetooth low energy built-in)	● Operates on 2.4 GHz ● 40 channels ● Power consumption: 0.01W to 0.5W

Table 1 - Controller Specs

2.5 Hardware Block Diagram

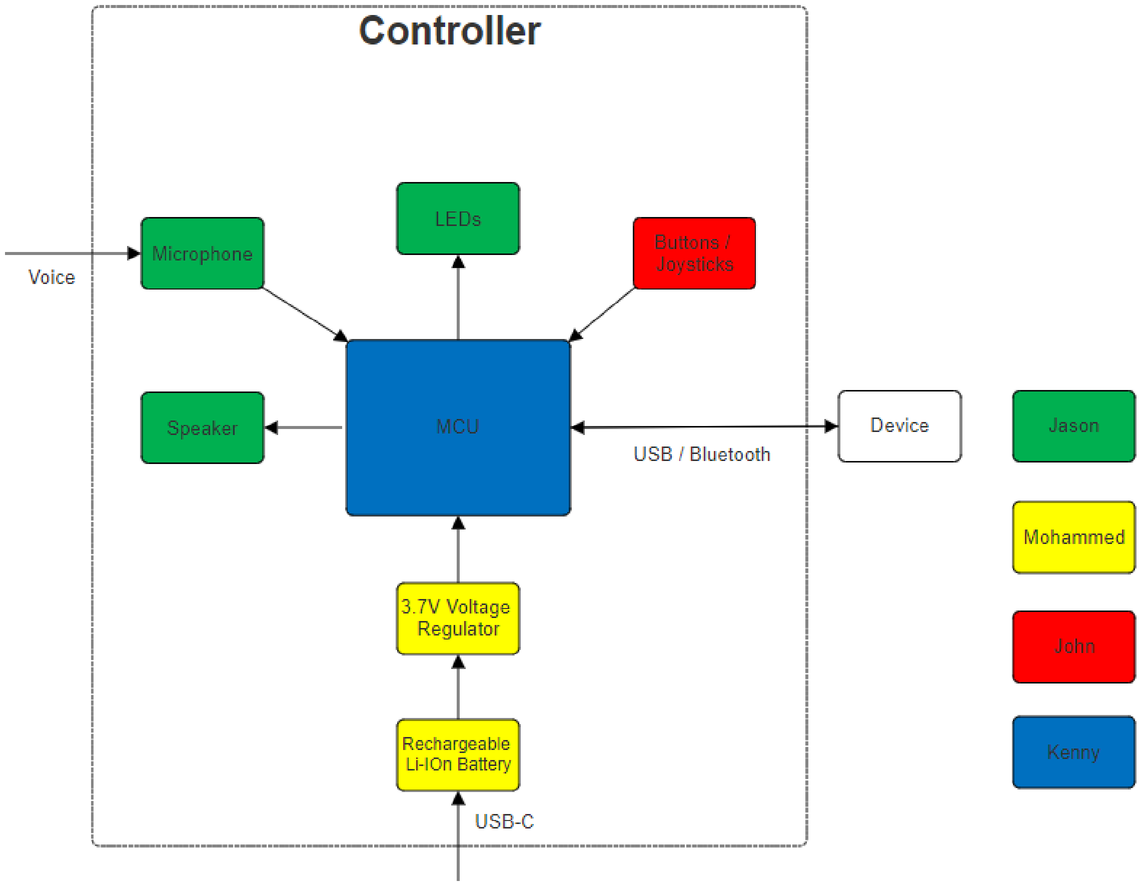

Figure 1 - Hardware Block Diagram

The buttons and inputs send user commands to the microcontroller, which processes these commands, manages power, communicates with the gaming device, and generates output signals as needed. The PCB serves as the structural and electrical platform for these components, while the batteries provide the necessary power for wireless operation. The gaming controller's relationship with the connected device is facilitated by the microcontroller's communication capabilities.

2.6 Software Diagram

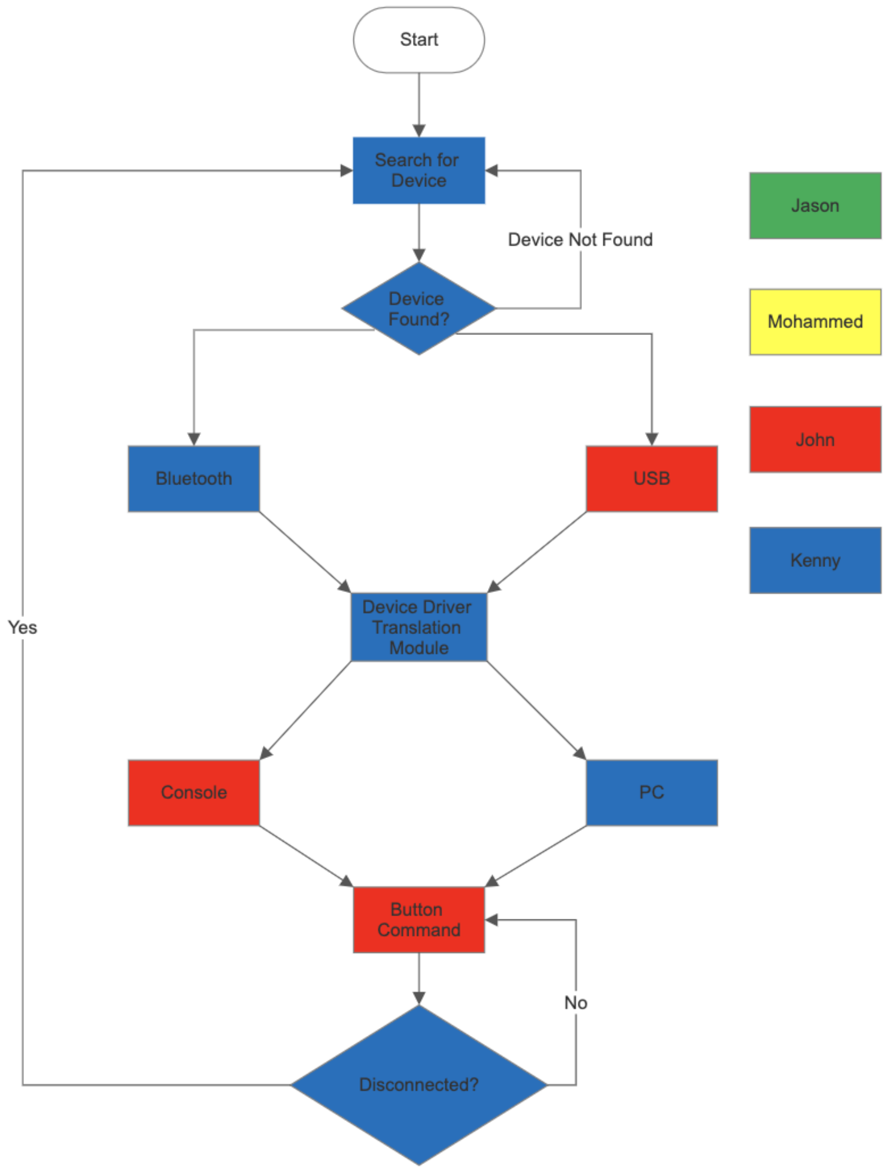

Figure 2 - Software Block Diagram

The software flow starts with initializing the hardware and establishing a Bluetooth connection or wired connection with the gaming device. The controller will detect and select the drivers for its respective device. The main program loop continuously reads and processes user input, updates the controller's state and logic, sends data to the connected device, and receives feedback or updates. It periodically checks for disconnect or end conditions, and upon detection, it disconnects, cleans up resources, and ends the software execution, ensuring seamless interaction between the controller and the gaming device through Bluetooth communication.

3.0 Research Related to Project Definition

Gaming transcends the boundaries of different platforms, devices, and operating systems. The need for a versatile and all-encompassing gaming controller has never been more evident. Gamers today demand a controller that seamlessly bridges the gap between consoles, PCs, mobile devices, and beyond, offering a unified gaming experience regardless of the hardware in use. At the same time, the persistent issue of joystick "stick drift" has plagued gamers worldwide, calling for innovative solutions to enhance the longevity and precision of gaming peripherals.

As we delve into the depths of this project, we will explore existing products and initiatives that have taken steps towards universality in gaming controllers. These products, each with its unique approach and target audience, offer valuable insights and lessons that can guide our own endeavor to create the ultimate universal gaming controller. Additionally, we will investigate the technological landscape surrounding hall effect sensors and their potential to revolutionize joystick durability and precision.

3.1 Existing Similar Projects and Products

This comprehensive analysis seeks not only to uncover the state of the art in universal gaming controllers but also to provide a blueprint for the design and development of a controller that embodies compatibility, innovation, and durability. Through an exploration of existing projects, and technologies, we aim to lay the groundwork for a universal gaming controller that truly transcends the barriers of gaming platforms and sets a new standard for immersive gameplay experiences.

3.1.1 Xbox Series X Controller

Figure 3 - Xbox Controller

We find ourselves drawing inspiration from the distinguished design and unparalleled compatibility of the Xbox Series X controller. This controller, a product of Microsoft's innovation, has not only set the bar for gaming controllers but has also been a pioneer in bridging the gap between consoles and personal computers.

The Xbox Series X controller's design is nothing short of exemplary. Its ergonomic shape and meticulously crafted buttons have made it a favorite among gamers worldwide. Its textured grips and well-placed controls ensure comfort during long gaming sessions. This dedication to ergonomic design serves as a blueprint for our own controller project, emphasizing the importance of user comfort and intuitive button layout.

One of the groundbreaking aspects of the Xbox Series X controller is its ability to seamlessly connect and function on multiple platforms. It was one of the pioneering controllers to extend its compatibility beyond the Xbox ecosystem. Players could enjoy the same controller experience on their PCs, thanks to Microsoft's dedication to driver support and Bluetooth connectivity **[3.1.1a]**. This cross-platform compatibility is a testament to the controller's versatility and has opened up new dimensions in gaming. While Xbox controllers have been available for PC gaming for an extended period, PlayStation controllers only recently ventured into the realm of PC compatibility. This distinction highlights the Xbox Series X controller's early and consistent commitment to universality.

However, one persistent drawback has been the susceptibility of its analog joysticks to stick drift over time **[3.1.1b]**. Another limitation is its connectivity, as it can only connect seamlessly to a specific number of devices within the Microsoft ecosystem.

3.1.2 8BitDo Ultimate Controller

Figure 4 - 8BitDo Ultimate Controller

The 8BitDo Ultimate Controller, a creation of 8BitDo, a company known for its commitment to versatility, stands out as a remarkable example of a controller designed to cater to a wide spectrum of gaming platforms.

The 8BitDo Ultimate Controller excels in ergonomic design, offering a comfortable grip and intuitive button layout that accommodates a broad range of hand sizes similar to Microsoft's controller.

The hallmark feature of the 8BitDo Ultimate Controller is its extraordinary compatibility. While it pays homage to classic controllers, it seamlessly integrates with various gaming platforms. This controller, through its Bluetooth connectivity, can be paired with Windows PCs, Android devices, iOS devices, and even the Nintendo Switch, making it an embodiment of universality **[3.1.2a]**. The controller's dedication to supporting multiple systems underscores its position as a go-to choice for gamers who enjoy a diverse gaming experience.

8BitDo goes the extra mile by offering customization options through firmware updates. This forward-thinking approach empowers users to tailor their controller experience to suit their individual gaming preferences. This aspect of user personalization aligns with our project's commitment to delivering a customizable gaming controller.

Compatibility-wise, the 8BitDo Ultimate Controller is at the top of the game. The controller does lack support for two of the biggest console giants in the market: Sony and Microsoft. What we would like to improve on is full controller support for Microsoft and Sony products and the addition of stick drift proof joysticks.

3.1.3 8BitDo USB Wireless Adapter 2

Figure 5 - 8BitDo USB Wireless Adapter 2

The 8BitDo USB Wireless Adapter 2 is a remarkable gaming accessory that brings versatility and compatibility to the forefront of the gaming experience. This device packs a powerful punch in terms of expanding controller options and streamlining cross-platform gaming.

At its core, the 8BitDo USB Wireless Adapter 2 is designed to bridge the gap between gaming controllers and different gaming platforms. It achieves this by allowing an extensive range of controllers, including those from the Nintendo Switch, PlayStation, and Xbox ecosystems, to connect seamlessly to a diverse array of devices. Its ease of use and simplicity make it a game-changer for gamers seeking a unified experience, regardless of the platform they choose to play on.

The 8BitDo USB Wireless Adapter 2 is characterized by its plug-and-play functionality, where users can effortlessly connect their preferred controller to their chosen device, whether it be a PC, a gaming console, or even certain mobile devices **[3.1.3a]**. This versatility empowers gamers to customize their gaming experience by using the controller they are most comfortable with, enhancing precision and control during gameplay.

Furthermore, the adapter's compact form factor ensures that it does not occupy excessive space or create clutter in a gaming setup. Its unobtrusive design allows users to discreetly integrate it into their gaming environment, making it a convenient and efficient solution for those who value both performance and aesthetics.

The 8BitDo USB Wireless Adapter 2 is an innovative third party piece of hardware that allows a wide array of controllers to be used on a wide array of devices. We admire their commitment to compatibility but chose the controller route instead of an adapter.

3.1.4 Comparison

	Xbox Series X Controller	8BitDo Ultimate Controller	8BitDo USB Wireless Adapter 2	Multipurpose Device Controller
Connectivity Type	Bluetooth, USB Type C	Bluetooth, USB Type C	Bluetooth	Bluetooth, USB Type C
System Compatibility	Xbox Series X, Xbox Series S, Xbox One, Windows 10/11, Android, and iOS	Nintendo Switch, SteamOS, Windows 10.	Switch 3.0.0, SteamOS, Windows 10, macOS, Raspberry Pi, Android TV Box, Retrofreak.	Xbox Series X, Xbox Series S, Xbox One, Windows 10/11, Playstation 4, Playstation 5, Nintendo Switch
Device Type	Controller	Controller	USB Adapter	Controller
Microphone	None	None	None	Yes
Speaker	None	None	None	Yes
Hall Effect Analog Sticks	None	None	None	Yes

Table 2 - Comparison to Other Devices

The Multipurpose Device Controller takes inspiration from several gaming devices to offer a versatile and adaptable gaming experience. It draws on the connectivity options of the Xbox Series X Controller and 8BitDo Ultimate Controller, providing both Bluetooth and USB Type C compatibility. It offers a wide system compatibility, encompassing Xbox consoles, Windows, PlayStation, and Nintendo Switch, similar to the Xbox Series X Controller and 8BitDo Ultimate Controller but extending its reach further. As a standard controller, it resembles the Xbox Series X Controller and 8BitDo Ultimate Controller, but it differentiates itself by incorporating a microphone for communication and voice commands. The controller also features a speaker, as seen in the other devices, and distinguishes itself by including Hall Effect Analog Sticks, enhancing precision and control. Overall, the Multipurpose Device Controller combines the best features of these devices to provide a comprehensive gaming solution for a variety of platforms.

3.2 Strategic Components and Part Selections

At the heart of our creation lies a carefully crafted blend of strategic components and meticulously curated part selections, all meticulously engineered to ensure seamless compatibility across a myriad of devices. Designed to adapt to your gaming needs, our controller embodies the pinnacle of connectivity, responsiveness, and precision, allowing you to conquer any platform with confidence and finesse.

3.2.1 USB-C Charging Interface

We have decided that having the universal serial bus USB-C charging interface on our gaming controller is one of our goals in this project. USB Type-C, often referred to simply as USB-C, is a versatile and universal connector technology for data and power transfer.. USB-C charging interface has a faster charging rate, it is more convenient and compatible for a huge number of devices. This makes the reliance on USB-C to charge our gaming controller reliable and efficient. We have taken into consideration many factors upon choosing the USB-C charging interface and what makes them reliable, the following will go into detail why this charging interface is the best choice by far for our gaming controller.

3.2.1.1 Universal Compatibility

USB-C or also known as the universal serial bus can be found nowadays with every other device as the group companies who developed this technology have under their membership over 700 companies, and not just any companies. It utilizes a compact, reversible, and symmetrical connector, which means there's no "upside-down" when plugging it in. USB-C employs a wide range of protocols, including USB 3.1, USB 3.2, Thunderbolt 3, and DisplayPort, making it a one-size-fits-all solution for various applications. Apple, Intel, Microsoft, and Samsung are among them. As a result, the consumer can benefit from this and use the interface to charge many devices with just one cable.

3.2.1.2 Data transfer Rates

The connector features multiple pins that handle data transfer, charging, audio, and video signals simultaneously. It provides faster data transfer speeds and higher power delivery (up to 100W), enabling rapid charging of devices and high-speed data transmission **[3.3.1.2a]**. The older generations of USB used to have a super speed of 5 Gigabytes per second for the single lane flow, and for the double lane flow the speed was 10 Gigabytes per second. The newer generation of USB 3.2 has doubled in the data flow exchange rate. As it went from 5 to 10 for the single lane flow. And from 10 to 20 Gigabytes per second for the double lane flow.

3.2.1.3 Charging Speed

As we mentioned earlier the cable of the UCB-C has two main jobs: charging and transferring data. Having a good reliable fast charger for any device is essential for customer service satisfaction. The UCB-C holds up to 12 watts of charging power for phones which is considered fast. And up to 240 watts of charging power for laptops depending on the cable and adapter.

3.2.1.4 Convenience

The design of the USB-C cables makes it easier for the consumer to use. It eliminates any frustration that the consumer might face with other cable options like the Micro USB, because it has the same design from both ends. Also, the design of one end makes the cable easier to plug, as there is no up or down orientation to it. These design features also expand the lifetime of the cable.

3.2.1.5 Comparison

We have considered many cables that would work on our gaming controller and the table goes through the option we had and highlights why we choose USB-C:

	Micro USB	USB-C	USB-A	Mini- USB
Compatibility	Bluetooth Headsets and Accessories, Smartphones and Tablets, Gaming Controllers	New laptops and phones, devices with combined data and power delivery. New Xbox gaming controller.	Flash drives, keyboards, and mice.	Older Digital Cameras, MP3 Players, GPS Devices.
Speed/ Data Transfer Rate For single cable	480 Mbps	480 Mbps, 5 Gbps, 10 Gbps	480 Mbps, 5 Gbps	480 Mbps
Design	Different ends/ different up/down orientation	Same ends/ same up/down orientation	Different ends/ different up/down orientation	Different ends/ different up/down orientation
Price	$5-$15	$10-$20	$5-$15	$5-$15

Version supported	2.0	2.0, 3.0 (3.1), 3.1 Gen 2	2.0, 3.0 (3.1), 3.1	2.0
Maximum Cable Length	16 feet	16 feet 2.0, 9 feet 3.0	16 feet 2.0, 9 feet 3.0	16 feet

Table 3 - USB Comparison

Opting for USB-C in a gaming controller proves superior for a contemporary gaming experience, primarily due to its reversible design, eliminating the hassle of proper orientation during connection. USB-C's higher data transfer rates ensure minimal input lag and swift communication between the controller and the gaming device, enhancing overall responsiveness. The increased bandwidth accommodates the transmission of complex data, such as high-resolution sensor inputs or advanced haptic feedback. USB-C's ability to deliver higher power levels facilitates fast charging for the controller's battery, minimizing downtime for gamers. In comparison to other USB protocols like USB-A and micro USB, USB-C stands out with its user-friendly design, faster data transfer rates, and enhanced power delivery capabilities, aligning perfectly with the demands of modern gamers for speed, convenience, and reliability in their gaming peripherals.

3.2.2 Batteries

Batteries are one of the most crucial components in our project. We had to look deep in this section because the choice of batteries in a gaming controller can impact many aspects of the gameplay and experience. In this section we will explore various aspects and types of batteries before choosing the right power source for our gaming controller.

3.2.2.1 Common Gaming controller Batteries

1. Lithium-ion (Li-ion) Rechargeable Batteries
2. Lithium-polymer (LiPo) Rechargeable Batteries
3. Nickel-Metal Hydride (NiMH) Rechargeable Batteries

3.2.2.2 Efficiency

It is the percentage of the output/input from the energy that goes through the battery, as there has to be energy losses during the process, this percentage will show us how efficient the batteries are. Lithium-ion batteries are the most efficient, the efficiency rate is over 99%. Second comes the LiPo with over 95% efficiency. Last on the list is Nickel-Metal Hydride with an efficiency rate of 80%.

3.2.2.3 Depth of Discharge

It is an indicator of the percentage of the battery that has been discharged relative to the overall capacity of the battery. Where depth of charge is the capacity that is discharged from a fully charged battery, divided by battery nominal capacity. For example, if we have 100Ah battery discharged for 20 minutes at current of 50A, DoD= 50*20/60/100=16.7%, which is the depth of discharge for this battery. Li-ion and Li-Po have a maximum discharge depth of 10-20%, both Li-Po and NiMH have a higher discharge depth of 20-30%.

3.2.2.4 Lifespan

Also known as the cycle of life, it is the number of discharge-charges that the battery goes through before it starts to lose its efficiency. Li-ion typical lifespan is around 300-500 discharge-charge cycles, Li-Po has the same rate of discharge-charge cycles. The NiMH varies from 200-300 discharge-charge cycles.

3.2.2.5 Power Density

A measure of how much power the battery can deliver related to its mass or size. This will allow us to qualify the batteries related to their size to know which one is better. So, the power density for NiMH is high, and it's measured around 1.5-4 Mega Watts per meter cube. For the lithium-ion and batteries it is slightly lower and measured around 0.4-2 MW/m cube. On the other hand, li-Po has higher power density because it's used in smaller applications that require more power.

3.2.2.6 Energy Density

A measure of how much energy available per unit volume or unit mass. And it is measured in Kilowatts hours per meter cube. First the NiMH has a 70-100 KWh/m cube, energy density is considered high related to other batteries. But the lithium ion and the lithium Polymer batteries have higher energy density than and they can deliver from 80 to 200 KWh/m cube.

3.2.2.7 Charge and Discharge Rate

A measure of how fast a battery can charge or discharge fully relative to its capacity. This is also known as C-rate, and it is measured in (mAh). 1C: Charging or discharging at the battery's rated capacity. For a 2000 mAh battery, 1C would be 2000 mA (2A). 0.5C: Charging or discharging at half the battery's rated capacity. For the same 2000 mAh battery, 0.5C would be 1000 mA (1A). 2C: Charging or discharging at twice the battery's rated capacity. For the 2000 mAh battery, 2C would be 4000 mA (4A). In this case if we have batteries with higher C it can charge up fast, but the downside to it is that it damages the batteries health and lifespan becomes shorter, so for both Li-ion and NiMH most of the time they both have the same rate which is 1C, and for li-Po it has a slightly higher rate of 2C.

3.2.2.8 Weight

Lithium-Po has less weight than both Li-ion and NiMH because as mentioned earlier it is used in smaller applications like phones. Then second comes Li-ion where they are considered less in weight as they don't have a metal component inside, lastly the NiMH has a higher weight among all three types.

3.2.2.9 Cost

Cost is not considered a major factor when choosing a gaming controller battery, because other factors affect the experience more. That being said, the cost of NiMH is less than 50% of a lithium battery in terms of the final battery pack production price. Also, lithium-Po are higher in price than regular li-ion batteries because they are used in phones, so they have to be more effective and less in weight and size.

3.2.2.10 Comparison

The below schedule highlights the choice for which will be going within our project. We have decided that lithium-ion batteries are the most suitable for our project.

	Lithium-ion	Lithium-polymer	Nickel-Metal Hydride
Efficiency	99%	95%	80%
DOD	Less	Less	More
Lifespan	300-500 cycles	300-500 cycles	200-300 cycles
Power density	Low	Higher	High
Energy density	High	Low	Low
C-rate	1C	2C	1C
weight	High	Low	Higher
Cost	Cheap	Expensive	Cheaper

Table 4 - Battery Comparison

3.2.3 Microcontroller Unit (MCU)

The selection of a robust Microcontroller Unit (MCU) is a pivotal aspect of our gaming controller project. The MCU serves as the central processing unit, responsible for interpreting user input, managing system functions, and facilitating seamless communication with gaming platforms and PCs. Our choice of MCU is backed by careful consideration of various factors that contribute to its reliability and efficiency. The MCU we decided to go with is the ESP32 due to its versatility and compatibility.

3.2.3.1 Processing Power and Efficiency

The heart of our gaming controller lies in its processors, specifically the Xtensa microprocessor. This dual-core (or single-core) 32-bit LX6 microprocessor operates at either 160 or 240 MHz, showcasing impressive performance metrics.

Single-Core Performance (240 MHz):

Operating at 240 MHz, the CPU achieves up to 600 DMIPS, demonstrating its computational prowess. Additionally, the inclusion of an Ultra Low Power (ULP) co-processor further enhances its capabilities.

Dual-Core Synergy (240 MHz):

In a dual-core configuration, each core operates at 240 MHz. This setup showcases exceptional performance metrics with 994.26 CoreMark and an impressive efficiency rating of 4.14 CoreMark/MHz. These processing capabilities empower our gaming controller to handle a diverse range of tasks with precision and efficiency. Whether it's rapid input processing or seamless communication with gaming platforms, the Xtensa microprocessor stands as a testament to our commitment to delivering a high-performance gaming experience.

3.2.3.2 Memory Capacity and Storage

The robust memory architecture of our gaming controller forms the cornerstone of its efficient data handling and processing capabilities. It comprises a diverse range of memory components, each meticulously selected to cater to specific tasks and operations.

Boot and Core Functions (ROM):
A substantial 448 KB of ROM is allocated for critical booting procedures and fundamental core functions. This dedicated space ensures the reliable initiation of the controller's operations.

On-Chip SRAM for Data and Instructions:
A generous 520 KB of on-chip SRAM is provisioned to handle both data and instructions, facilitating swift and seamless processing of information.

RTC FAST Memory (SRAM):
Within the Real-Time Clock (RTC), an additional 8 KB of SRAM, known as RTC FAST Memory, serves as a dynamic storage space. It is accessed by the main CPU during RTC boot from the Deep-sleep mode, contributing to efficient data management.

RTC SLOW Memory (SRAM):

Another 8 KB of SRAM, designated as RTC SLOW Memory, is available for access by the Ultra Low Power (ULP) co-processor during the Deep-sleep mode. This configuration enhances operational efficiency during low-power states.

External Flash and RAM Expansion:
The controller supports up to 16 MB of external flash memory, seamlessly mapped into CPU instruction and read-only memory spaces simultaneously. This expansion bolsters the available resources for program execution. Additionally, external RAM can be mapped into CPU data memory space. The controller accommodates SRAM up to 8 MB, with the capability to map up to 4 MB at a time. This flexibility in memory access, spanning 8-bit, 16-bit, and 32-bit reads and writes, enhances versatility in data handling.

This diverse array of memory resources ensures that our gaming controller can efficiently manage data, execute instructions, and handle diverse tasks with precision and speed.

3.2.3.3 Peripheral Integration

The versatile peripheral integration of our gaming controller extends its functionality to a wide array of operations, ensuring compatibility with various devices and systems. This comprehensive suite of peripherals enriches the controller's capabilities and facilitates seamless interactions with the gaming environment.

Touch Sensors:
A total of 10 touch sensors are integrated, allowing for intuitive and responsive touch-based interactions. This feature enhances the controller's usability and expands its potential for interactive gaming experiences.

Communication Interfaces:
The controller supports an extensive range of communication interfaces, including SPI, I2S, I2C, UART, and Bluetooth features. These interfaces are equipped to handle diverse communication protocols and can be configured to operate in various modes, offering versatility in connecting with external devices.

Wireless Connectivity with Bluetooth:
Bluetooth has seamlessly integrated itself into the realm of modern gaming controllers, with the ESP32's Bluetooth capabilities taking center stage. In the context of gaming, the ESP32's Bluetooth connectivity serves as the conduit for wireless communication between the gaming controller and the console or PC, liberating gamers from the entanglement of cumbersome wires. Operating within the 2.4 GHz frequency band, the ESP32 employs the frequency-hopping spread spectrum technique to transmit data between the controller and the gaming device. This involves rapid switching between various frequencies within the 2.4 GHz band, establishing a secure and interference-resistant connection **[3.3.3.3a]**. By synchronizing their hopping sequences, the ESP32-equipped gaming controller and the console create a robust link, ensuring a seamless gaming experience. The controller is equipped with Bluetooth capabilities, providing a seamless connection to compatible devices. It supports a UART HCI interface with speeds of up to 4 Mbps, an SDIO/SPI HCI interface, and a PCM/I2S audio interface for enhanced audio

experiences. The Bluetooth stack of the chip is compliant with the Bluetooth v4.2 BR/EDR and Bluetooth LE specifications.

Furthermore, the ESP32's Bluetooth technology boasts low power consumption, making it an energy-efficient choice for prolonged gaming sessions. Its capability to connect with multiple devices simultaneously enhances its versatility, facilitating communication not only with gaming consoles but also with smartphones and other compatible devices. With the ESP32's Bluetooth, players can relish a hassle-free gaming experience, free from input lag and characterized by a stable and reliable connection.

Storage and Networking Capabilities:
The controller features a host interface for SD/eMMC/SDIO, facilitating seamless data storage and retrieval. Additionally, an Ethernet MAC interface with dedicated DMA and IEEE 1588 support ensures efficient networking capabilities.

Additional Specialized Features:
The controller offers specialized functionalities such as Motor PWM, LED PWM with up to 16 channels, and compatibility with ISO 11898-1 (CAN Specification 2.0) via TWAI®.
The integration of these diverse peripherals empowers our gaming controller with the flexibility and adaptability required to provide a seamless and immersive gaming experience

3.2.3.4 Power Efficiency and Management

Efficient power utilization is at the core of our gaming controller's design, ensuring extended battery life and optimized performance. The controller employs a suite of fine-resolution power control techniques, allowing precise adjustment of clock frequency, duty cycle, Wi-Fi operating modes, and individual power states of internal components.

Diverse Power Modes:
Our controller offers five distinct power modes tailored to various usage scenarios: Active, Modem-sleep, Light-sleep, Deep-sleep, and Hibernation. These modes are engineered to strike a balance between performance and power consumption.

Deep-sleep Mode Efficiency:
Notably, the Deep-sleep mode exhibits remarkable power conservation, with consumption as low as 10 µA. This mode is particularly advantageous during extended periods of inactivity, ensuring minimal energy drain.

Ultra-Low-Power (ULP) Coprocessors:
The inclusion of ULP coprocessors further enhances power efficiency. These specialized components operate with minimal power requirements, augmenting the controller's ability to conserve energy during critical tasks.

Voltage Range and Supply Recommendations:
The controller operates within a voltage range of 2.3 V to 3.6 V. When employing a single-power supply configuration, it is recommended to utilize a power source delivering 3.3 V, with an

output current of 500 mA or higher. This voltage and current combination ensures stable and reliable operation. Efficient power management lies at the heart of our gaming controller's design philosophy. By implementing a diverse array of power modes and utilizing specialized components, we aim to maximize battery life and enhance the overall user experience.

3.2.3.5 Real-Time Processing Capabilities

The real-time processing capabilities of our gaming controller are facilitated by a comprehensive suite of clock sources, timers, and watchdogs. These features ensure precise timing, robust fault recovery, and seamless transitions between power modes.

Clock Sources:
The controller incorporates a range of clock sources to drive its operations. These include an internal 8 MHz oscillator with calibration, an internal RC oscillator with calibration, an external 2 MHz to 60 MHz crystal oscillator (with 40 MHz dedicated to Wi-Fi/Bluetooth functionality), and an external 32 kHz crystal oscillator for the Real-Time Clock (RTC) with calibration.
Upon reset, an external crystal clock source is selected as the default CPU clock. This external crystal clock source also interfaces with a Phase-Locked Loop (PLL) to generate a high-frequency clock, typically operating at 160 MHz. Additionally, the controller features an internal 8 MHz oscillator.

RTC Clock Sources:
The Real-Time Clock (RTC) module is driven by five potential clock sources, providing flexibility in various operating scenarios. These sources include an external low-speed (32 kHz) crystal clock, an external crystal clock divided by 4, an internal RC oscillator (typically around 150 kHz, and adjustable), an internal 8 MHz oscillator, and an internal 31.25 kHz clock (derived from the internal 8 MHz oscillator, divided by 256).
Selection of the clock source depends on the power mode and CPU accessing requirements. In normal power mode with a need for faster CPU accessing, the application can opt for the external high-speed crystal clock divided by 4 or the internal 8 MHz oscillator. In low-power mode, the choices include the external low-speed (32 kHz) crystal clock, the internal RC clock, or the internal 31.25 kHz clock.

3.2.3.6 Reliability and Robustness

The ESP32 is engineered to deliver exceptional reliability, even in demanding industrial environments. Its operational capacity spans a wide temperature range from –40°C to +125°C, ensuring stable performance across diverse conditions. To augment its robustness, the ESP32 is equipped with sophisticated calibration circuitries that actively mitigate external circuit imperfections and seamlessly adapt to variations in external conditions.

Advanced Error Counters:
The ESP32 integrates advanced error counters that diligently monitor system behavior, enabling prompt identification and resolution of anomalies. These counters serve as a vital tool for tracking and managing errors effectively.

Configurable Error Interrupt Threshold:
The controller features a configurable error interrupt threshold, allowing for customized error handling strategies. This adaptable threshold empowers users to fine-tune error detection sensitivity based on specific operational requirements.

Error Code Capture:
In the event of an error occurrence, the ESP32 is equipped with the capability to capture error codes. This feature aids in diagnosing the root cause of errors, streamlining the debugging process and expediting resolution.

Arbitration Lost Capture:
The controller incorporates an arbitration lost capture mechanism, ensuring graceful recovery from arbitration challenges. This feature plays a crucial role in maintaining system integrity and reliability, particularly in scenarios where multiple processes vie for access to shared resources. The combination of these features fortifies the ESP32's reliability and robustness, making it well-suited for deployment in industrial settings with stringent operational requirements.

3.2.3.7 Compatibility with Customization Features

The ESP32's innate versatility extends to its compatibility with customization features, offering extensive flexibility for tailored configurations. Post-initialization, the firmware gains the capability to dynamically remap external RAM or flash into the CPU's address space, providing a powerful tool for expanded button mapping options.

This customization capability empowers developers to optimize the controller's functionality to suit specific gaming preferences and requirements. By strategically allocating resources within the CPU's address space, the controller can seamlessly adapt to diverse gaming scenarios, enhancing the overall user experience.

The ESP32's inherent compatibility with customization features underscores its adaptability and empowers developers to unlock the full potential of the gaming controller, delivering a personalized and immersive gaming experience.

3.2.3.8 Driver Development

The ESP32 is a particularly advantageous choice for driver development due to several key factors. First and foremost, the ESP32 is equipped with a powerful dual-core processor, which provides ample computational resources for handling driver-related tasks efficiently. This processing capability is crucial for managing various peripherals and ensuring responsive communication with external hardware.

Additionally, the ESP32 comes with a rich set of built-in peripherals and interfaces, including GPIO pins, I2C, SPI, UART, and more. This extensive feature set simplifies driver development by providing direct access to these peripherals, reducing the need for complex external circuitry and making it easier to interface with a wide range of sensors, actuators, and other devices.

Espressif's ESP-IDF (Espressif IoT Development Framework) is another compelling reason for choosing the ESP32. The IDF is a comprehensive and well-documented framework that includes a variety of APIs and libraries specifically designed for the ESP32. These resources streamline the development process, providing standardized methods for configuring and interacting with different peripherals. The IDF's modular structure also contributes to code organization, making it easier to create and maintain drivers.

Furthermore, the ESP32 community is vibrant and actively contributes to the development ecosystem. This means that there is a wealth of online resources, forums, and community support available for developers working on driver development. Collaborating with this community can provide valuable insights, troubleshooting assistance, and even opportunities for code contributions.

3.2.3.9 Comparisons

The below table highlights the choice for which will be going within our project. We have decided that the ESP32 is the most suitable for our project.

	ESP32	STM32WB15CC	RX23W	MSP430FR5989
Memory	448 KB ROM, 520 KB of SRAM, 8 KB of SRAM RTC FAST, 8 KB of SRAM RTC SLOW Memory	320 KB flash memory, 48 KB SRAM	384 - 512 KB ROM 64 KB RAM	128 KB ROM 2 KB RAM
Processing Power	dual-core (or single-core) 32-bit LX6 microprocessor operates at either 160 or 240 MHz	frequency up to 64 MHz, MPU, 80 DMIPS and DSP instructions	Maximum operating frequency: 54 MHz	16 MHz

Peripherals	34 programmable General Purpose Input/Output (GPIO), 12-bit Successive Approximation Register (SAR), two 8-bit Digital-to-Analog Converter (DAC), SPI, I2S, I2C, UART, and Bluetooth features	Bluetooth® Low Energy, 1x DMA controller (7x channels) supporting ADC, SPI, I2C, USART, AES, timers, 1x LPUART (low power) 1x SPI 32 Mbit/s 1x I2C (SMBus/PMBus ®)	Simple I2C Simple SPI MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPCK (RSPI clock) enables serial transfer through SPI operation (four lines) Bluetooth low energy (BLE)	2 x UART 48 x GPIO 12 x ADC channels 2 x I2C 4 x SPI 5 x timers
Power Management	five distinct power modes: Active, Modem-sleep, Light-sleep, Deep-sleep, and Hibernation. Voltage range of 2.3 V to 3.6 V	1.71 to 3.6 V power supply, 610 nA Standby mode + RTC + 48 KB RAM Active-mode MCU: 33 µA	Operation from a single 1.8-V to 3.6-V supply Three low power consumption modes	Active Mode: Approximately 100 µA/MHz Standby (LPM3 With VLO): 0.4µA Real-Time Clock (RTC) (LPM3.5): 0.35µA Shutdown (LPM4.5): 0.02µA
Cost	$9.99 USD	$2.70 USD	$9.42 USD	$11.48 USD

Table 5 - MCU Comparison

Designing a gaming controller with the ESP32 microcontroller proves to be a great choice for several reasons. Its integrated Wi-Fi and Bluetooth capabilities align seamlessly with the demand for wireless gaming experiences, enabling the controller to communicate effortlessly with gaming consoles or PCs. The dual-core processor provides robust processing power, ensuring the responsiveness required for complex gaming inputs, and the architecture supports efficient multitasking. With an array of peripherals including touch sensors, GPIO pins, and analog-to-digital converters, the ESP32 offers versatility for implementing a variety of gaming controller features such as responsive buttons and analog sticks.

Moreover, the ESP32's low power consumption is critical for battery-powered devices, enhancing the longevity of the gaming controller during extended play sessions. The extensive support from the ESP32 community and readily available documentation simplify the development process, providing a wealth of resources for troubleshooting and customization. Cost-effectiveness further sweetens the deal, making the ESP32 an attractive option for hobbyists and small-scale projects. Additionally, the ESP32's open-source nature fosters innovation, allowing developers to modify the firmware to meet specific project requirements. In essence, the ESP32 stands out as a well-rounded choice, combining wireless capabilities, processing power, versatility, energy efficiency, community support, cost-effectiveness, and open-source flexibility for an optimal gaming controller design.

3.2.4 Analog Stick Precision and Performance

A Hall effect joystick is a type of input device that employs Hall effect sensors to determine the position of a joystick's handle or lever. It operates based on the Hall effect, which is the phenomenon where the voltage across a conductor changes when exposed to a magnetic field. Inside the joystick, there are typically two or more Hall effect sensors placed at right angles to each other. When the user moves the joystick, it repositions a magnet embedded in the joystick's handle. As the magnet moves, it generates a magnetic field that is sensed by the Hall effect sensors. By measuring the changes in voltage generated by these sensors, the device can accurately determine the position of the joystick in two dimensions (X and Y) and sometimes even the Z-axis, offering precise control for gaming, robotics, or other applications **[3.3.4a]**. Hall effect joysticks are known for their durability and accuracy, making them valuable components in various industries.

The analog sticks in our gaming controller are engineered to provide unparalleled precision and performance, offering gamers a highly responsive and immersive gaming experience. This section will delve into the key attributes that make these analog sticks a pivotal component of our controller's design.

3.2.4.1 High Resolution and Sensitivity

Our analog sticks boast a high level of resolution and sensitivity, allowing for fine-grained control over in-game movements. This precision ensures that gamers can execute even the most delicate maneuvers with accuracy, enhancing their gameplay experience.

3.2.4.2 Dynamic Range and Dead Zone Calibration

The analog sticks in our gaming controller are equipped with customizable dynamic range. This capability empowers users to finely adjust the responsiveness of the sticks, creating a personalized gaming experience that aligns precisely with their individual preferences.

In contrast to analog sticks, Hall Effect sticks exhibit significantly reduced wear and are far less prone to producing drift over time. Consequently, controllers employing Hall Effect sticks require minimal dead zone calibration to account for potential drift, if any.

This translates to a notably heightened level of responsiveness with Hall Effect sticks. Even subtle movements of the stick are accurately reflected on-screen, offering a potential competitive edge, particularly in fast-paced genres like shooters.

3.2.4.3 Ergonomic Design for Comfortable Gameplay

The design of the analog sticks prioritizes ergonomics, ensuring that gamers can comfortably navigate through extended gaming sessions. The sticks are sculpted to fit naturally within the player's hand, reducing fatigue and enabling sustained gameplay without discomfort.

3.2.4.4 Durability and Longevity

Our analog sticks are meticulously crafted from high-quality materials and subjected to rigorous testing, ensuring they are built to last. Designed to withstand the demands of even the most intense gaming sessions, these sticks offer reliable performance over an extended period.

Analog joysticks, also known as 'potentiometer' joysticks, operate by manipulating components that physically interact with each other, thus altering the electrical resistance of the circuit. This mechanism allows for highly precise tracking, as it involves measurable changes in electrical resistance. However, it's important to note that over time, the resistors may experience wear due to the physical contact between components.

In contrast, Hall Effect joysticks offer a different approach. They do not suffer from the traditional drift commonly observed in analog joysticks. Instead, they utilize magnets within Hall Effect sensors. While it's possible for these magnets to experience changes or loss of magnetism, the likelihood of such occurrences is significantly lower compared to analog stick drift. To induce alterations in magnetism, Hall Effect joysticks would need to be exposed to substantial levels of electromagnetic interference.

3.2.4.5 Dual Analog Stick Configuration for Enhanced Gameplay

The dual analog stick configuration further enhances gameplay possibilities, providing a greater range of motion and enabling more complex interactions within the gaming environment. This configuration opens up new avenues for immersive gaming experiences.

3.2.4.6 Comparisons

The below table highlights the choice for which we will be going within our project. We have decided that the hall effect is the most suitable for our project.

	Hall Effect Joystick	**Joystick**
Longevity	over 10 million cycles	500,000 - 1,000,000 cycles
Rotary Mechanism	magnets and an electrical conductor	Potentiometers

Axies	3	2
Cost	$5.10	$3.52

Table 6 - Joystick Comparison

Opting for a Hall effect joystick in gaming controllers offers advantages rooted in its magnetic field-based, contactless design. This not only ensures increased durability and longevity, crucial for gaming enthusiasts, but also eliminates issues of friction and mechanical wear. The absence of physical contacts in Hall effect joysticks translates to a more stable and reliable gaming control, reducing the likelihood of input jitter or drift.

Additionally, Hall effect joysticks provide precise and accurate sensing, enhancing responsiveness for a more immersive gaming experience. The joystick's customizable sensitivity settings further contribute to user comfort, allowing gamers to fine-tune the response to match their preferences. The Hall effect joystick's contactless, durable, and precise characteristics make it a superior choice for gaming controllers, providing a reliable, long-lasting, and responsive solution for gamers seeking enhanced control and accuracy in their gaming experience.

3.2.5 Action Buttons: Intuitive Gameplay Control

8 action buttons are strategically positioned on the controller's face, providing quick and intuitive interactions for gamers. These buttons play a pivotal role in executing crucial in-game actions, including jumping, shooting, and performing special moves, along with being used for the directional pad(up, down, left, and right). The selection of these components was carefully made, focusing on obtaining cost-effective buttons in packs that offer the best value. These same buttons will also be used for our shoulder buttons.

3.2.5.1 Button Type and Functionality

Our action buttons are designed as digital inputs, ensuring precise and responsive feedback during gameplay. Their intuitive placement allows for seamless execution of various in-game commands, enhancing the overall gaming experience.

3.2.5.2 Electrical Specifications

Rated Voltage: Min - 1 V dc, Max - 12 V dc
Rated Current: Min - 0.01 mA, Max - 50 mA
Withstanding Voltage for 1 Minute: 250 V ac
Contact Resistance: Under a load of 2 times the operating force at 5 V dc, 10 mA - 100 mΩ

3.2.5.3 Durability and Reliability

Constructed with high-quality materials and subjected to rigorous testing, our action buttons are engineered for long-lasting durability. They are designed to withstand the demands of extended gaming sessions, ensuring reliable performance over time.

3.2.5.4 Ergonomic Design for Comfortable Gameplay

The action buttons are ergonomically sculpted to provide comfort during prolonged gaming sessions. Their tactile feedback and responsive click ensure a satisfying and enjoyable gameplay experience.

3.2.5.5 Comparisons

	TS04-66-50	TL3305CF260QG
Life Span	80,000 Cycles	200,000 Cycles
Switch Travel distance	0.25mm	0.20mm
Operating Force	160gf	260gf
Mounting Type	surface	surface
Cost	$0.18	$0.26

Table 7 - Button Comparison

The TL3305CF260QG buttons stand out as an excellent choice for gaming controllers, primarily due to their tact switch design that provides tactile and responsive feedback, crucial for gaming precision. This tactile feel ensures confident button presses, enhancing the overall gaming experience. Additionally, these buttons are designed for durability, with a high actuation rating, making them well-suited for the intense and frequent usage characteristic of gaming sessions. Their compact and surface-mount design further facilitates easy integration into the compact form factor of gaming controllers, offering a reliable, tactile, and durable solution for crafting controllers that prioritize precision, responsiveness, and long-term performance.

3.2.6 Dynamic Speaker

Our gaming controller is equipped with a dynamic speaker designed to deliver an immersive audio experience, enhancing gameplay interactions and providing clear, dynamic sound.

3.2.6.1 Type of Speaker

The speaker incorporated in our controller is a dynamic speaker, known for its ability to produce high-quality sound with excellent clarity and range.

3.2.6.2 Specifications

Weight: Approx. 3.0 grams
Magnet Material: Nd-Fe-B
Impedance: 8 ± 15% Ohm at 1000Hz, 1V
Power Rating: Normal - 1.0W, Maximum - 1.5W
Frequency Output (FO): 670 ± 20% Hz
Sound Pressure Level (S.P.L.): 96 ± 3dB/1.0Watt at 0.1 Meter. Average at 1000Hz, 1200Hz, 1500Hz, 2000Hz.
Frequency Range: F0~20,000Hz. Average SPL-10dB. Frequency Response Fig (3).
Distortion: 10% Maximum at 1000Hz, 1.0W.
Abnormal Sound Test: Must be normal tested by 2.83 Volts. Sine wave for 50~5kHz
Operating Temperature: -20°C to +55 °C
Storage Temperature: -25 °C to +65 °C

3.2.6.3 Superior Sound Quality

The dynamic speaker integrated into our controller is engineered to provide superior sound quality, ensuring that players receive an immersive audio experience that complements their gaming sessions.

3.2.6.4 Comparisons

	SP-3605 90DB	4227 96DB
Frequency Range	300 Hz ~ 5 kHz	670 Hz ~ 20 kHz
Self resonant range	550Hz	670Hz
Impedance	8 Ohms	8 Ohms
Decibels	90.00	96.00
Max Power	1.5 W	1.5 W
Cost	$2.90	$1.95

Table 8 - Speaker Comparison

The 4227 96DB speaker is a superior choice for the gaming controller due to its higher decibel rating of 96.00, promising a louder and more immersive audio experience compared to the SP-3605's 90.00. Additionally, the 4227 boasts a wider frequency range of 670 Hz to 20 kHz, providing better reproduction of diverse audio frequencies essential for a dynamic and realistic gaming atmosphere. Despite a slightly higher cost at $1.95, the enhanced audio capabilities make the 4227 96DB a more compelling option for delivering superior sound performance in the gaming controller.

3.2.7 Integrated Microphone

Our gaming controller comes equipped with an integrated microphone designed to deliver crystal clear audio, enhancing communication during multiplayer sessions and ensuring a seamless gaming experience.

3.2.7.1 Universal Compatibility

The integrated microphone is designed to work seamlessly with a wide range of devices, providing versatile compatibility for various gaming platforms and communication applications.

3.2.7.2 Microphone Specifications

Directivity Omnidirectional
Sensitivity (S) at 1 kHz (0 dB = 1 V/Pa) Min: -38 Max:-36
Supply Voltage (VDD) Min: 1.0 V Max: 10 V
Current Consumption (VDD = 2.0 V, RL = 2.2 kΩ) Max: 0.5 mA
Sensitivity Reduction (VDD = 2.0 ~ 1.5 V) 3 dB
Frequency (f) Min: 100 Hz Max: 10,000 Hz
Signal to Noise Ratio (S/N) at 1 kHz, Pin = 1 Pa (A-weighted) 60 dBA
Total Harmonic Distortion (THD) at 94 dB SPL, 1 kHz 1%
Total Harmonic Distortion (THD) at 115 dB SPL, 1 kHz 3%
Output Impedance (Zout) at 1 kHz 2.2 kΩ

3.2.7.3 Superior Sound Quality

The integrated microphone is engineered to provide superior sound quality, ensuring clear and effective communication between players during gaming sessions.

3.2.7.4 Comparisons

	MEMS Microphone	Electret Condenser Microphones
Frequency	36 Hz to 20 KHz	100-10,000 Hz
Signal to Noise Ratio	63 dBA	60 dBA
Sensitivity	-32 dBV FS	-33 dB
Cost	$0.62	$1.44

Table 9 - Microphone Comparison

The MEMS microphone is the superior choice for the gaming controller, offering a wider frequency range of 36 Hz to 20 kHz compared to the Electret Condenser Microphone's 100-10,000 Hz. Despite its lower cost at $0.62, the MEMS microphone provides a higher signal-to-noise ratio of 63 dBA, ensuring cleaner and clearer audio for a more immersive gaming experience.

3.2.8 High-Performance Switch

Our gaming controller features a high-performance switch meticulously designed to provide precise control and seamless interaction, enhancing the gaming experience for players.

3.2.8.1 Universal Compatibility

The high-performance switch is engineered to seamlessly integrate with a wide range of gaming platforms, ensuring compatibility with various gaming systems and devices.

3.2.8.2 Switch Specifications

Contact Rating: 0.3A @ 6VDC or 0.1A @ 30VDC
Electrical Life: 5,000 make-and-break cycles
Contact Resistance: 70 mΩ max.
Insulation Resistance: 100 MΩ min. @ 500V
Dielectric Strength: 500 VAC min. @ sea level
Operating Temperature: -40ºC to +85ºC
Storage Temperature: -40ºC to +85ºC

3.2.8.3 Reliability and Durability

Crafted with high-quality materials and subjected to rigorous testing, our high-performance switch is built for reliability and durability, ensuring it can withstand the demands of prolonged gaming sessions.

3.2.8.4 Comparisons

	Sub-Miniature Slide Switches	TI Analog Switch ICs Single Bilateral Analog Switch
Mounting style	PCB Mount	SMD/SMT
Current Rating	300 mA	100 mA
Voltage rating (DC)	6 V	5.5 V Max
Resistance	70 Ohms	10 Ohm
Cost	$0.59	$0.94

Table 10 - Switch Comparison

The Sub-Miniature Slide Switch is the optimal choice for our gaming controller, featuring a higher current rating of 300 mA, a superior voltage rating of 6 V (DC), and lower resistance at 70 Ohms, all at a more cost-effective price of $0.59 compared to the TI Analog Switch ICs at $0.94.

3.2.9 LEDs

Our project aims to have LEDs to make the experience of the user more evolving and fun. In order to improve the experience, we decided to include LEDs that have certain tasks depending on the state of the controller. The LEDs will show when the power is on or when turning off, it will also provide a certain color when it reaches a power charge below a certain percentage in our batteries.

3.2.9.1 Size and Shape

For a gaming controller the LEDs included must be mini size, in the range where it will be able to put them inside the controller or on top.

3.2.9.2 Color

Static color modes are important, there is no specific color count, but we do have a minimum while exploring our options, at least 8 color modes will be enough to achieve the tasks we need to interact with the user.

3.2.9.3 Efficiency

Efficiency is an important factor in choosing an LED for our gaming controller. Efficiency can include power consumption, brightness and intensity controls, visual impact, customization, and software control. First power consumption, the LEDs we plan to use in our gaming controller will have a lion share of energy in our MCU as the MCU is going to rely on the 3.7 volts coming from the Lithium-ion batteries which would be enough to supply all the components except for the LEDs. LEDs will typically consume 5-24 volts, so in our choice we want to consider an LED with good power consumption and also good efficacy. Software controllers for the LED we are going to use is going to be important as well for the efficiency as some LEDs don't come with a set up configuration which will make it easy for the project as a whole if we have a ready configuration to be used. Brightness level is crucial whether the controller is being used in a room that is dark or full of gaming lights. Visibility, and constant color production are important.

3.2.9.5 Power

The input power of the LEDs we want to include in our gaming controller has to match the voltage of the MCU we are using, because no other component needs voltage regulation, so the LEDs will be taking power from the MCU. The MCU will operate on the Lithium-ion batteries which supply around 3.7 volts. And there is going to be a step-up voltage circuit to ensure the LEDs are working. The ideal voltage intake will be 5-12 volts.

3.2.9.6 Comparison

	Adafruit 5050 RGB LED with Integrated Driver Chip	WL-SFTW SMT Full-color TOP LED
Voltage	5 V	5 V
Current	5 mA	20mA Red, 20mA Green, 20mA Blue
Power	10 watts	108 mW (Blue) 108 mW(Green) 72 mW(Red)
Dimensions LxWxH	5.12 x 3.15 x 0.39 inches	3.50mm x 2.80mm 2.10mm
Cost	6.99 (10)	$0.50 (1)

Table 11 - RGB Comparison

The Adafruit 5050 RGB LED is the superior choice for the gaming controller LED despite a slightly higher cost of $6.99 for a set of 10. It outperforms the WL-SFTW SMT LED with lower power consumption at 10 watts compared to WL-SFTW's combined 288 mW and features integrated control. Additionally, the Adafruit LED's larger dimensions contribute to a more visually appealing lighting effect.

3.2.10 Power Bank

Our gaming controller project incorporates a power bank to ensure uninterrupted gameplay and provide a reliable power source for the controller's components. The power bank serves as a portable energy reservoir, enhancing the controller's usability and extending its operational duration. We are in between power sources, with the ease of use of the power bank and the versatility of a battery pack that we will go into further detail later.

3.2.10.1 Capacity

The power bank boasts a substantial capacity of 10000mAh or 37Wh, providing ample energy to sustain prolonged gaming sessions. This capacity aligns with the power demands of the gaming controller, ensuring a reliable and enduring power source.

3.2.10.2 Input and Output

The power bank features efficient power management with an input capability of 20W (max) and output capability of 22.5W (max). This dual functionality allows the power bank to swiftly recharge and simultaneously power the gaming controller, accommodating the demands of both input and output seamlessly.

3.2.10.3 Size and Weight

Designed with portability in mind, the power bank's dimensions are 4.15 x 2.60 x 0.94 inches (105 x 66 x 24mm), making it compact and suitable for integration into the gaming controller setup. With a weight of 6.88 ounces (195g), the power bank maintains a lightweight profile, contributing to the overall ergonomic design of the gaming controller.

3.2.10.4 Efficiency and Voltage Compatibility

Efficiency is a paramount consideration in selecting the power bank for our gaming controller project. The power bank aligns with the voltage requirements of the gaming controller's MCU (ESP32 WROOM-32E), which operates on the 3.7 volts supplied by Lithium-ion batteries. The power bank's output voltage is compatible with the operational needs of the gaming controller without the need for additional voltage regulation.

In conclusion, the power bank integrated into our gaming controller project acts as a reliable and efficient energy source, ensuring extended gameplay without interruptions. In the end we decided on the bank with the lower weight and smaller size so it could be more easily integrated into our design.

3.2.10.5 Comparison

	INIU Portable Charger, Small 10000mAh Power Bank USB C in/Output 22.5W	**Vida IT Small Power Bank 5V 2A Battery Pack for Heated Vest Jacket Gloves 5000mAh Dual USB**
Capacity	10000mAh	5000mAh
Input	20W(max)	5V, 2.1A
Output	22.5W(max)	5V, 2.1A
Size	4.15*2.60*0.94in	3.82 x 2.44 x 0.59 inches
Weight	195g	126g

Table 12 - Power Bank Comparison

3.2.11 Battery Pack

In navigating the power landscape for our gaming controller project, we find ourselves at the crossroads of power bank and battery pack options. Each comes with unique advantages, and the battery pack, in particular, warrants careful consideration due to its distinct characteristics.

3.2.11.1 Voltage and Capacity

The battery pack operates at a voltage of 3.7V, aligning with the power requirements of the gaming controller's MCU (ESP32 WROOM-32E). While offering a typical capacity of 820mAh and a minimum capacity of 770mAh, the battery pack provides a compact yet reliable power source. This capacity, though smaller than the power bank, can be advantageous in scenarios where minimizing weight and size is a priority.

3.2.11.2 Weight and Size

Weighing approximately 16.4g and boasting dimensions of 30.5 x 44 x 6.8mm (1.2" x 1.69" x 0.26"), the battery pack epitomizes portability. Its compact size makes it a compelling option for integration into the gaming controller, allowing for efficient use of space without compromising on power.

3.2.11.3 Comparison with Power Bank

In direct comparison to the power bank, the battery pack offers a trade-off between capacity and size. While it may not match the power bank's extended capacity of 10000mAh, the battery pack's diminutive size and weight make it an attractive alternative, especially in situations where space and weight constraints are critical considerations.

3.2.11.4 Efficiency and Integration

Efficiency remains a key factor in evaluating the battery pack for our gaming controller. Its voltage aligns with the requirements of the ESP32 MCU, eliminating the need for additional voltage regulation. The efficient use of space and the ability to integrate seamlessly into the gaming controller design enhance the overall appeal of the battery pack.

In conclusion, the battery pack emerges as a compelling option in our quest for an optimal power solution. Its balance of voltage, capacity, weight, and size makes it a viable alternative, particularly in scenarios prioritizing portability and space efficiency. As we navigate the intricate design of our gaming controller, the battery pack stands as a versatile contender in the power landscape.

3.2.11.5 Comparison

	EEMB 3.7V Lithium ion Battery 820mAh 653042 Rechargeable 3.7 Volt Lipo Battery Pack	3.7V 1800mAh Lithium Polymer LiPo Rechargeable Battery
Voltage	3.7V	3.7V
Capacity	820mAh	1800mAh

Weight	16.4g	16.3g
Size	1.2"x1.69"x0.26" (WxLxH)	8mm (H)*34mm (W)*50mm (L)±2mm

Table 13 - Battery Pack Comparison

3.2.12 Voltage Regulator

As we delve into the intricacies of our gaming controller's power management system, the role of a voltage regulator becomes paramount. A vital component in ensuring stable power delivery, the chosen voltage regulator comes with specific features that align with the nuanced requirements of our design.

3.2.12.1 Type and Specifications

The selected voltage regulator is a DC-DC buck converter, offering a versatile solution for our gaming controller's power distribution needs. Its specifications include:

Input Voltage: DC 3.2V - 35V
Output Voltage: DC 1.25V - 30V (adjustable)
Output Current: 3A (max.)
Output Power: 15W (max.)
Conversion Efficiency: 92% (max.)
Switching Frequency: 65KHz
Rectification: Non-synchronous rectification
Module Property: Non-isolated buck module
Operating Temperature: -45℃ to +85℃
Load Regulation: ± 0.5%
Voltage Regulation: ± 2.5%
Size (L x W x H): 45 x 23 x 14mm / 1.77 x 0.91 x 0.55inch
Weight: 14g
3.2.12.2 Purpose and Integration

The voltage regulator plays a pivotal role in maintaining a consistent voltage output for our gaming controller's components. With an adjustable output voltage ranging from 1.25V to 30V, it accommodates the diverse voltage requirements of various elements within the circuit. This adaptability ensures that sensitive components receive the precise voltage needed for optimal performance.

3.2.12.3 Efficiency and Stability

Boasting a remarkable maximum conversion efficiency of 92%, this voltage regulator stands as a reliable and efficient solution. The non-isolated buck module design, combined with non-synchronous rectification, contributes to minimizing power loss and enhancing overall stability in the power supply chain.

3.2.12.4 Size and Temperature Tolerance

Designed with compact dimensions of 45 x 23 x 14mm and weighing a mere 14g, the voltage regulator is tailored for integration into our gaming controller without compromising on space efficiency. Its operational temperature range from -45℃ to +85℃ ensures robust performance across varying environmental conditions.

3.2.12.5 Comparison

	LM2596 DC to DC High Efficiency Voltage Regulator 3.2-35V	**5V-5.3V 5.2V 3.5-6A Transformer Dual Output Voltage Regulator**
Input Voltage	3.2V - 35V	3V-36 V
Output Voltage	1.25V - 30V (adjustable)	5-5.3V
Output Power	15W (max.)	25W
Switching frequency	65KHz	NA
Size	1.77 x 0.91 x 0.55inch	63 x 27 x 10mm

Table 14 - Voltage Regulator Comparison

3.3 Possible Architectures and Related Diagrams

To craft the finest multipurpose device controller, we began by studying existing controller designs and versions used for device management. In this section, we look into various types of controllers and how they're structured to achieve their intended purposes. Our primary focus centers on gaming controllers since our device caters primarily to gaming needs. However, it's worth noting that our controller is designed to go beyond gaming, acting as a versatile tool for managing an array of devices, making it a fundamental component in a wide range of digital interactions**.** We will begin with a list of different gaming architectures used for a variety of tasks.

3.3.1 Traditional Gamepad

Traditional gamepads, like those used for consoles like Xbox and PlayStation, have a well-defined architecture that integrates hardware and software components to provide a seamless gaming experience. These gaming controllers feature a range of button and input elements, including thumbsticks for precise analog control, triggers on the top shoulders for actions like shooting and acceleration, a D-pad for digital input, pressure-sensitive action buttons, start and select buttons for in-game navigation, a platform-specific home button for system control, and additional buttons for complex input. The controllers prioritize ergonomics with comfortable grips, well-placed buttons, and textured surfaces for a secure hold. They offer both wired (USB) and wireless (Bluetooth or proprietary) connectivity options, with wireless models powered by rechargeable or disposable batteries. Gamepad software controls haptic

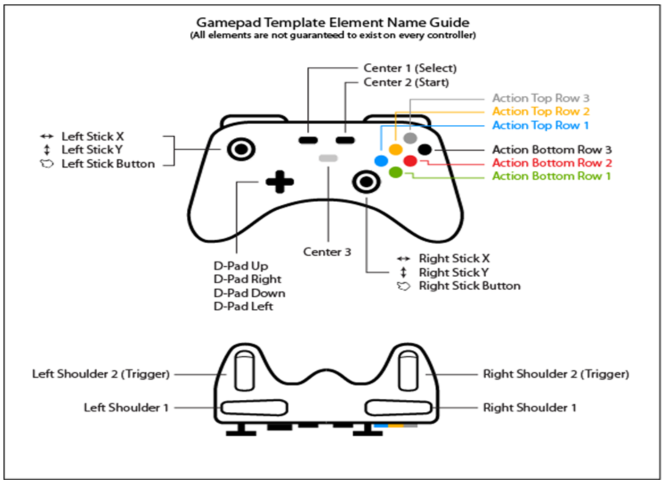

Figure 6 - Gamepad Template

feedback (vibration), enhancing immersion by providing tactile responses to in-game events like gunfire, collisions, or environmental effects.

3.3.2 Arcade Stick (Joystick)

Arcade stick controllers are defined by their robust hardware design, consisting of a central joystick with a sturdy, spring-loaded shaft for precise control, detected via sensors like potentiometers or optical encoders. Accompanying the joystick are large, responsive buttons strategically placed to mimic classic arcade machine control panels, engineered to endure rapid and forceful presses during intense gameplay. The entire assembly is encased in a rectangular enclosure, designed to mirror the aesthetics of classic arcade cabinets while maintaining stability with a weighted baseplate. These controllers typically connect to gaming platforms through USB or proprietary connectors, with detachable cables for portability. In terms of software architecture, arcade stick controllers maintain a focus on straightforward input detection, offering users the ability to remap buttons to suit their preferences or game requirements. Ensuring compatibility across various gaming platforms and achieving minimal input lag are paramount, with optimizations for swift and accurate translation of joystick movements into on-screen actions.

Figure 7 - Commodore Controller

3.3.3 Motion Controllers

Motion controllers are equipped with a diverse array of sensors to track their movement and orientation, including accelerometers for detecting velocity and acceleration changes, gyroscopes to measure angular velocity and orientation, magnetometers for orientation tracking and drift reduction, and infrared sensors or cameras for precise tracking through external markers or beacons. In addition to these motion sensors, these controllers typically feature standard buttons, thumbsticks, and trigger buttons for added interactivity. Many motion controllers also incorporate haptic feedback mechanisms, such as vibration motors or force feedback, to deliver tactile sensations in response to in-game actions. Their ergonomic designs are crafted for comfort, with variations tailored to specific applications, like gun-shaped controllers for shooting games. Connectivity-wise, they wirelessly connect to gaming platforms or VR systems through Bluetooth or other wireless protocols, with power supplied via USB charging or disposable batteries depending on the model.

Figure 8 - Meta Quest 2

On the software front, motion controllers rely on sensor fusion techniques to amalgamate data from their sensors, like accelerometers and gyroscopes, ensuring precise tracking while mitigating drift. Input mapping within the software translates sensor data and button inputs into commands recognized by gaming platforms and VR applications, offering customization for different gaming experiences. Low latency is a paramount focus, ensuring that in-game actions mirror the controller's movements without noticeable delay. These controllers offer diverse interactivity, recognizing gestures, object interactions, and in-game events driven by the user's actions. Haptic feedback control is an integral feature, providing tactile responses in the form of vibrations or force feedback to enhance immersion. Calibration and configuration settings enable users to fine-tune sensitivity, dead zones, and other parameters according to their preferences. Compatibility with specific gaming platforms or VR systems is also assured through software integration, guaranteeing seamless recognition and support across various games and applications.

3.3.4 Gaming Keyboards

Gaming keyboards are specialized input devices designed to meet the unique demands of gamers. They distinguish themselves from standard keyboards by offering features such as customizable key lighting, dedicated macro keys, anti-ghosting, and mechanical key switches. In terms of hardware design, gaming keyboards incorporate mechanical key switches, durable keycaps, RGB backlighting for visual customization, and varying form factors to suit different preferences. They may include dedicated macro keys, additional controls, wrist rests, USB pass-through ports, and robust build quality. Software architecture plays a crucial role in managing key assignments, RGB lighting, and advanced functionalities, allowing users to customize key functions, lighting profiles, and macros. Game Mode features prevent accidental key presses, firmware updates enhance performance, and profile management allows users to adapt the keyboard for different games. Gaming keyboards often integrate with a broader gaming ecosystem and may offer cloud synchronization for profile and settings accessibility on multiple devices. This combination of hardware and software delivers a responsive, customizable, and immersive gaming experience, making gaming keyboards indispensable for competitive and enthusiast gamers.

Figure 9 - Keyboard

3.3.6 Dongle Architecture

The design option we considered for our multipurpose controller involved a wireless gamepad solution based on LPC5528 and NxH3670. The LPC5528 microcontroller, with its Cortex-M33 architecture, clocking in at a main frequency of up to 150 MHz, and boasting a flash size of up to 512 KB and a RAM size of up to 256 KB, presented itself as a compelling host controller. It was equipped with two USB device controllers, offering both Full-speed USB and High-speed USB support, a vital characteristic for gamepad applications. On the other hand, the NxH3670 played the role of an ultra-low-power 2.4 G wireless transceiver, integrating a Cortex-M0 processor and

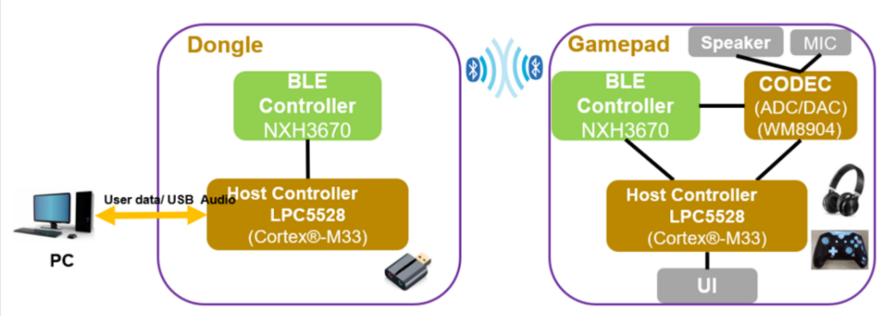

Figure 1. Wireless gamepad solution based on LPC5528 and NxH3670

Figure 10 - Dongle Architecture

being certified for Bluetooth Low Energy 4.2. Notably, the NxH3670 offered low latency and minimal power consumption, attributes that are highly desirable in wireless gaming headset products.

However, after careful consideration, our team opted not to pursue the dongle architecture. While this setup held significant promise, it posed some limitations that influenced our decision. These limitations included potential complexity in the deployment and management of dongles, the need for additional hardware components, and the potential for interference with other wireless devices. We also recognized the growing trend towards seamless, integrated connectivity solutions, which led us to favor the ESP32 with Bluetooth technology for its simplicity, efficiency, and wide compatibility across multiple devices and platforms. Ultimately, our choice of the ESP32 was guided by our goal of creating a versatile, user-friendly, and highly functional multipurpose controller.

3.3.7 ESP32 Architecture

Our team's selection of the ESP32 microcontroller for our multipurpose controller was grounded in its remarkable capabilities and an array of features that made it the perfect fit for our project. The ESP32 module, whether in the form of ESP32-MINI-1 with an on-board PCB antenna or the ESP32-MINI-1U with an external antenna connector, showcased significant strengths. Both versions were equipped with a 4 MB flash in a chip package, ensuring ample storage capacity for firmware and applications.

One of the pivotal factors that led us to choose the ESP32 was its robust power management. The 5 V to 3.3 V low-dropout (LDO) power regulator seamlessly converted the power supply from 5 V to 3.3 V, ensuring stable and efficient operation. The board featured a Boot Button for firmware download mode initiation, enhancing the ease of updating firmware via the serial port. Additionally, the Reset Button provided a straightforward means of restarting the microcontroller when needed. The presence of a Micro-USB Port offered dual functionality, serving as both the power supply interface for the board and the communication interface between a computer and the ESP32 chip. This versatile design streamlined the connectivity and operation of our multipurpose controller. The USB-to-UART Bridge, a single USB-UART bridge chip with transfer rates of up to 3 Mbps, ensured rapid and reliable data exchange.

The 3.3 V Power On LED indicator provided a visual cue of the board's operational status when connected to the USB. Lastly, the I/O Connector was a crucial element that facilitated our project's adaptability. It exposed all available GPIO pins (with the exception of the SPI bus for flash) through pin headers on the board. This configuration empowered us to program the ESP32 chip for multiple functions, enhancing the controller's versatility.

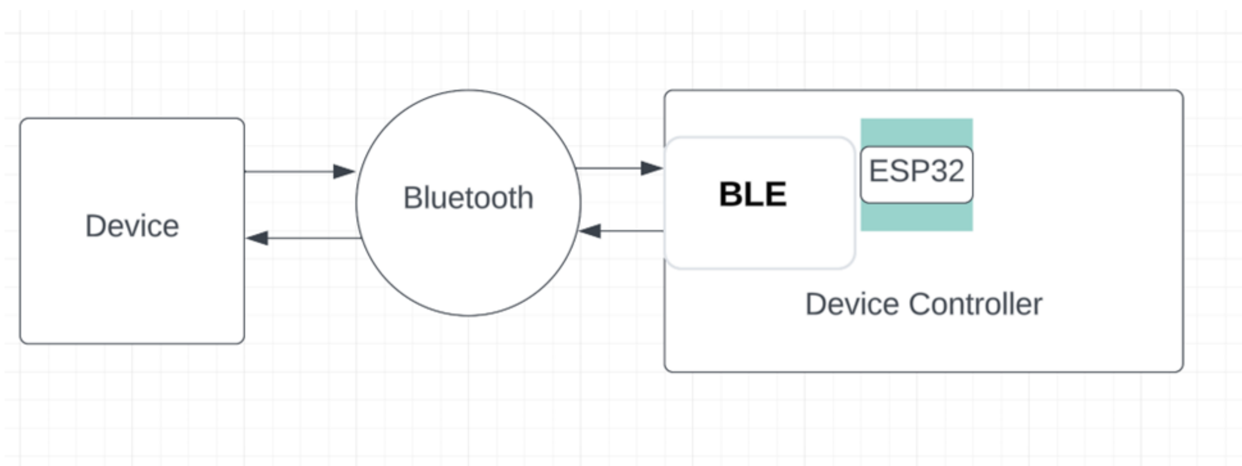

Figure 11 - Bluetooth Diagram

3.3.8 Architectural Design Choice

Our team explored two distinct approaches for connecting our multipurpose controller. The first approach involved using a dongle as an intermediary device to establish connections with various platforms and devices. The second, which emerged as the preferred choice, centered around harnessing the capabilities of the ESP32 microcontroller as our primary communication interface, employing Bluetooth technology.

The decision to adopt the ESP32 was rooted in a range of practical and advantageous factors that made it the ideal fit for our multipurpose controller. The ESP32 offered an elegant solution for wireless connectivity, eliminating the need for additional hardware like dongles. Moreover, its extensive compatibility extended to a diverse array of devices, ranging from gaming platforms to

PCs and mobile devices. This expansive compatibility simplified the user experience, allowing the controller to seamlessly interact with a wide spectrum of digital environments.

Crucially, the ESP32's energy-efficient design and capability for low-latency communication aligned perfectly with our vision for prolonged battery life – a pivotal consideration for both gaming and extended device control scenarios. Its open-source nature and thriving development community ensured ongoing adaptability and continuous improvement of the controller's features and capabilities through regular software updates and enhancements.

The dual-core architecture of the ESP32, coupled with its robust processing capabilities, was instrumental in minimizing input lag, thereby ensuring that the controller offered a smooth and responsive user experience. Furthermore, the microcontroller's multitasking prowess enabled it to effectively manage a gamut of tasks, from processing sensor data to interpreting user inputs, thus enhancing the overall performance of our multipurpose controller. Its innate support for secure communication further bolstered data integrity and user privacy, bolstering our commitment to delivering a trustworthy and dependable product.

In summary, the ESP32's selection as the primary communication interface in our multipurpose controller was driven by its capacity to offer seamless wireless connectivity, wide device compatibility, low-energy consumption, adaptability, and robust processing capabilities. These features collectively contribute to the creation of a versatile, responsive, and secure controller, adept at meeting the diverse needs of gamers and device users while accommodating future improvements and developments in the dynamic realm of digital interactions.

In summary, the ESP32 microcontroller emerged as the ideal choice for our multipurpose controller due to its efficient power management, ease of firmware updates, versatile USB interface, and ample GPIO pins for extensive functionality. These attributes collectively contributed to the development of a highly adaptable and user-friendly multipurpose controller that met the diverse needs of our project.

4.0 Related Standards and Realistic Design Constraints

Designing a gaming controller is an exciting endeavor that involves a careful balance of creativity, innovation, and adherence to established standards and design constraints. It's crucial to recognize that the gaming industry, with its ever-evolving technology and consumer demands, demands the highest level of precision. Creating a successful gaming controller requires a comprehensive understanding of industry standards governing ergonomics, button layouts, connectivity, and safety, among other factors. Equally important is the recognition of realistic design constraints, such as budget, manufacturing capabilities, and compliance with regulatory requirements. We will explore the essential standards and practical limitations that will guide the development of a game controller that not only delights gamers but also meets the expectations of a dynamic and competitive market.

4.1 Related Standards

It's essential to consider the myriad of standards that govern this dynamic and rapidly evolving industry. Gaming controllers are the bridge between players and virtual worlds, and to ensure seamless interaction, adherence to established standards is paramount. These standards encompass a wide range of aspects, from ergonomic considerations to button layout conventions, wireless communication protocols, and regulatory compliance. Understanding and embracing these standards is not only a mark of professionalism but also a guarantee of compatibility and safety. We will delve into the critical standards that will shape the design process and result in a product that not only meets industry expectations but also provides an exceptional gaming experience for users worldwide.

4.1.1 USB-C Standards

USB-C, short for Universal Serial Bus Type-C, represents a modern and versatile standard for connectivity in the digital world. Known for its small, reversible connector and high data transfer speeds, USB-C has become the go-to choice for a wide range of devices, from laptops and smartphones to external storage and peripherals. The USB-C standard encompasses several crucial features, including support for faster data transfer rates (up to 10 Gbps in USB 3.1 Gen 2) **[4.1.1a]**, power delivery, which allows for the charging of devices and even laptops, and the ability to transmit audio and video signals. Furthermore, USB-C is designed to be universal, making it easy to connect and transfer data across a broad spectrum of devices, irrespective of the manufacturer or operating system. This adaptability and speed have made USB-C a pivotal player in the technology ecosystem, driving the evolution of digital connectivity in our increasingly interconnected world.

4.1.2 C++ Standards

C++ standards are pivotal in defining the evolution of the C++ programming language. These standards, established and maintained by the International Organization for Standardization (ISO), continually shape the landscape of C++ development. Beginning with the original C++98 standard, subsequent iterations such as C++03, C++11, C++14, C++17, and C++20 introduced a wealth of new features and improvements to the language **[4.1.2a]**. These standards serve as a foundation for ensuring code portability across different compilers and platforms, promoting consistency in the language's usage, and enabling developers to harness the power of modern C++ features. The standards not only provide a roadmap for language development but also offer guidance on best practices and design principles, making C++ a robust and adaptable language for a wide range of application domains, from system programming to game development and beyond.

4.1.3 Electromagnetic Compatibility Standards

Electromagnetic Compatibility (EMC) standards are a vital set of guidelines and regulations that address the electromagnetic interference (EMI) produced by electronic and electrical devices. These standards are essential to ensure that different electronic equipment and systems can coexist harmoniously without causing interference to each other or experiencing susceptibility to external disturbances. EMC standards set limits on the levels of electromagnetic emissions and establish criteria for immunity to electromagnetic disturbances. Compliance with these standards is crucial for manufacturers, as it guarantees the reliability and safety of electronic devices, prevents potential interference issues, and ensures that products can function as intended within a complex and interconnected world of electronics. EMC standards are applied in a wide range of industries, from consumer electronics to automotive, aerospace, and medical devices, emphasizing the importance of managing electromagnetic disturbances for the benefit of both manufacturers and end-users **[4.1.3a]**.

4.1.4 Power Standards

Compliance with power standards is crucial to ensure the efficient and reliable operation of the gaming controller. This subsection delves into the impact of voltage and current requirements on the controller's design, considering the specific power standards associated with gaming consoles:
USB-C Charging Interface: The selection of the USB-C charging interface is underpinned by its ability to support faster charging rates and offer universal compatibility. It's important to note that all USB-C cables must be capable of carrying a minimum of 3A current at 5V (equivalent to 15W), although some cables can handle up to 5A current at 20V (equivalent to 100W). This information guides the controller's power management system.

Xbox One Controllers: These controllers require a voltage input ranging from 2.4 to 3.0 volts. Understanding this range is vital in ensuring compatibility with Xbox gaming systems.
DualShock 4 Controller: The DualShock 4 controller operates optimally at a power rating of 5V and draws a current of 800mA (0.8A) for normal operation and charging. This specification serves as a critical reference point for designing the power circuitry.

ESP32 Operating Voltage Range: The ESP32 microcontroller, a central component of the gaming controller, operates within a voltage range of 2.2 to 3.6V. In typical operation, the ESP32 Thing supplies power to the chip at 3.3V, aligning with its specified operating range.
Understanding and adhering to these power standards is foundational to the design process, ensuring seamless compatibility with gaming platforms and reliable performance across various scenarios.4.1.2.2

4.1.5 Bluetooth Standards

Bluetooth protocols form the backbone of the controller's wireless communication capabilities. This subsection delves into how Bluetooth standards influence the design, with a focus on ensuring seamless connectivity with a wide range of devices, including gaming consoles, PCs, and mobile platforms. Adhering to established Bluetooth protocols is instrumental in enhancing the controller's versatility.

Frequency Band: Bluetooth communication operates within specific frequency bands. The relevant range is between 2.402 and 2.480 GHz, or 2.400 and 2.4835 GHz, with guard bands of 2 MHz at the lower end and 3.5 MHz at the upper end. These frequencies fall within the globally unlicensed, yet regulated, industrial, scientific, and medical (ISM) 2.4 GHz short-range radio frequency band. This knowledge informs the controller's wireless communication design.
Bluetooth 5.0 Advancements: Bluetooth 5.0 introduces a significant enhancement by increasing the bandwidth to 2 Mbps. This doubling of data transfer capacity results in reduced transmission times for devices, enabling swift and reliable over-the-air firmware updates for mobile devices. Additionally, it facilitates the speedy upload of large datasets collected from sensors when a mobile device is powered on. Understanding and implementing these Bluetooth standards is pivotal in ensuring robust wireless communication capabilities for the controller, facilitating seamless interactions with a diverse array of platforms and devices.

4.1.6 UART and SPI Standards

UART and SPI standards play a pivotal role in facilitating serial communication, allowing seamless interaction with various peripherals. This subsection delineates the profound impact that adherence to these standards has on the controller's capability to exchange data with crucial components like sensors and displays. It emphasizes the advantages of compatibility and interoperability.

UART Communication: In UART communication, the maximum data rates typically range from 230 kbps to 460 kbps. To ensure accurate data transmission, both the transmitting and receiving devices within the UART system must be configured to employ the same data format and baud rate (the number of bits transmitted per second). This adherence guarantees reliable and efficient data exchange.

SPI Communication: SPI (Serial Peripheral Interface) is a synchronous communication protocol reliant on a clock signal to synchronize data exchange between devices. It finds extensive application in scenarios demanding higher data transfer speeds and support for multiple devices, such as memory chips, display controllers, and sensor networks. Operating in a master/slave configuration, the master device governs communication with multiple slave devices. It initiates data exchange using individual chip select lines and generates the clock signal. SPI facilitates full-duplex communication at exceptionally high speeds. The SPI bus encompasses four essential signals or pins:
Master – Out / Slave – In (MOSI)
Master – In / Slave – Out (MISO)
Serial Clock (SCLK)
Chip Select (CS) or Slave Select (SS)
Understanding and adhering to these UART and SPI standards significantly augment the controller's capacity for seamless data exchange with an array of vital peripherals.

4.2 Realistic Design Constraints

Designing a gaming controller is a complex task that requires careful consideration of various realistic design constraints. These constraints shape the controller's form, functionality, and overall usability. From the need to create an ergonomically comfortable device for gamers to ensuring compatibility across multiple gaming platforms, and optimizing power efficiency for extended gameplay, this introduction explores the multifaceted challenges faced by designers in crafting a successful gaming controller. We will delve into these design constraints to better understand their impact on the development and user experience of gaming controllers.

4.2.1 Economic and Time Constraints

The development of the gaming controller is subject to various economic and time-related constraints. This section provides a comprehensive exploration of the practical limitations that shape the design process:

4.2.1.1 Budget Allocation

Economic constraints play a pivotal role in resource allocation for the project. Thorough research and assessment were conducted to identify components that not only met our project's requirements but also fell within our allocated budget. This subsection delves into the budgetary considerations that impact critical aspects such as component selection, manufacturing processes, and overall production costs.

We meticulously scoured the market to find the best prices while ensuring each component fits seamlessly into our project's design. Our budget allocation for each part reflects a careful balance between cost-effectiveness and quality. While we wanted to get the best possible components for our project, we had to keep in mind that the best is not always the most expensive. Components like the Microcontroller, batteries, and buttons all have multiple iterations that specialize in different aspects; so deciding which were the most suitable along with the most price effective was something we took our time to consider.

4.2.1.2 Development Timeline

Time constraints are a critical aspect of project management. This subsection examines the challenges posed by deadlines and outlines the methodology for efficient project scheduling and task prioritization.

Our initial milestones involve defining the project's specifications, objectives, and constraints. We meticulously finalized the detailed requirements for the gaming controller, taking into account critical components like BLE integration, analog joysticks, trigger buttons, batteries, USB-C port, speakers, mic, on/off switch, D-pad buttons, and more. Additionally, we established a clear budget, ensuring that component costs were well-balanced with quality considerations.

Next, using Autodesk Eagle, we embarked on creating a detailed schematic for the gaming controller. This served as the blueprint for the hardware components and their interconnections. Even with the user-friendly interface of Autodesk Eagle, creating the schematic that is suitable for our project and makes logical sense will be another time constraint to consider. Finding a similar schematic online and modifying it can save time in the future.

Concurrently, we delved into code development, focusing on joystick calibration, button functionality, and Bluetooth connectivity. Seamless integration with various devices was paramount during this phase. The Bluetooth connectivity may be the most time consuming part along with developing drivers so the controller can properly interface with the connected device. As we progressed into the second semester, we turned our attention to ordering the necessary components for the gaming controller. Careful consideration of component prices and alternatives was essential to stay within budget. Given the nature of global sourcing, PCB manufacturing and shipping times introduced a significant constraint. We meticulously factored in these lead times to ensure they aligned with our project schedule.

With components in hand, we consider the assembly process, meticulously soldering the PCB, integrating the batteries, and connecting various hardware elements. Attention to detail is paramount to ensure proper connections and functionality, because having to order additional parts due to human error will cost us more time in the end when testing should be our focus. Fine-tuning joystick calibration and thorough testing of all controller functions followed. This phase includes extensive refinement of code to enhance Bluetooth and USB-C connectivity. Involving potential users or testers provided invaluable feedback on the gaming controller's performance and user experience.

In the final weeks of the second semester, comprehensive testing will be conducted to validate functionality, assess battery life, and evaluate overall performance. Stress testing will be implemented to ensure the controller meets its intended usage requirements. Our timeline reflects the pragmatic considerations we undertook, acknowledging the constraints of time, budget, and global sourcing. This process ensures that our project was not only innovative but also realistic and feasible within the allocated resources and expertise of our team

4.2.2 Environmental, Social, and Political Constraints

Designing a gaming controller doesn't exist in a vacuum; it is subject to a web of interconnected environmental, social, and political constraints. This introduction sets the stage for a comprehensive exploration of the intricate interplay between these three sets of constraints, highlighting the pivotal role they play in the design and acceptance of gaming controllers.

4.2.2.1 Environmental Constraints

Our gaming controller is considered as an electronic device so as many other electronics devices have environmental constraints our gaming controller would have a supply chain to other electronics. Any electronic devices need mining first to extract all the elements out of the earth. Mining some elements that would be used in a gaming controller like gold, copper, lead, nickel, zinc, lithium, cobalt and cadmium requires huge amounts of energy to separate these elements in specialized factors. The process also requires huge amounts of water as well. As we all know, the production of power can be harmful to the environment as most of the production is used by fossil fuels like oil, natural gas, and coal. These sources release greenhouse emissions into the air which leads to an increase in the world global temperature.

4.2.2.2 Social Constraints

A big part of social constraints would be accessibility to gaming controllers and playing games in general. Not all individuals have the capability to buy or use a gaming controller. For example, there are so many people with disabilities which make the current designs or the gaming controller itself as an object hard to use by those people. That being said, there has been more research and development in the past few years to make gaming controllers accessible even to the people who have disabilities of any sort to make the gaming controller and gameplay experience more comprehensive and accessible for everyone.

4.2.2.3 Political Constraints

There are many political constraints on gaming controllers and video games in general. Constraints may vary from country to country depending on the rules and regulations of each culture. First, content regulation, the content of any game depends on the developers and who is going to play the game, as some games may contain kill scenes or shooting which, the gaming controller will enhance the experience of these scenes by having some reactions like vibration when shooting or when the character is dead. The problem that faces this situation is allowing such a game to be played in a certain region as some cultures have constraints on what they want their people to consume in regard to video games in general.

4.2.3 Ethical, Health, and Safety Constraints

In the dynamic world of gaming controller design, a multifaceted set of ethical, health, and safety constraints must be carefully considered. This section illuminates the intricate landscape of ethical, health, and safety constraints that are increasingly crucial in the modern gaming industry, framing the discourse on how designers navigate these complex issues to create responsible and user-centric gaming controllers.

4.2.3.1 Ethical Constraints

When dealing with ethical constraints, gaming controllers might have some issues. As they have been controversial ever since they have been first launched. As some would say gaming controllers have proven to be addictive, and as any other addiction this addiction can be harmful to people’s lives. The design of the controller itself can affect the brain as it provides comfort, the comfort will lead to longer gaming sessions that last for ours which then will turn to be addiction. Also, the responsiveness of gaming controllers and the constant feedback whether it was a voice, or a vibration enhance the brain to react which will make the gameplay experience leads to addiction eventually. Moreover, developers usually design the game and include many addicting elements in the game play. For example in a fighting game, the more the costumer plays the game there will be progress into some levels, the levels this person will reach will face new challenges that requires some weapons that might include within game purchase which and if someone played the game for so long the probability that they are going to spend money on the game makes the user even more attached to the game they are playing. All these elements cause addiction to the gameplay.

4.2.3.2 Health

Health constraints on gaming controller usually include the material that the gaming controller is made of, as we all know most of the gaming controllers uses plastic as a cover, the use of plastic poses some problem as much research indicated that microplastic can cause health problem not only for the environment and animals, but for humans as well. In case any of these microplastics particles get in the digestive system of human beings they might cause short to long term health issues to a person.

4.2.4 Manufacturing and Sustainability

For the successful large-scale production of our multipurpose device controller, we need to navigate several manufacturability and sustainability constraints, particularly in the context of modern challenges like chip shortages and overseas PCB production. These constraints have a direct impact on the efficiency, cost-effectiveness, and environmental sustainability of our production process.

4.2.4.1 Supply Chain Resilience

The ongoing chip shortage highlights the importance of securing a stable supply of critical components, like microcontrollers, essential for our multipurpose device. This may involve diversifying suppliers, maintaining buffer stock, and establishing close relationships with key chip providers to adapt to shifting market conditions.

4.2.4.2 Sustainability and International PCB Production

To address sustainability concerns related to international PCB production, we need to explore options like local or regional PCB suppliers. This can reduce the carbon footprint associated with shipping and offer a more sustainable approach to sourcing essential components for our device. Additionally, opting for energy-efficient production methods, sustainable materials, eco-friendly packaging, and efficient recycling processes can enhance the environmental sustainability of our manufacturing and product.

4.2.4.3 Quality and Scalability

Maintaining high standards of quality control, rigorous testing, and efficient production line layouts are pivotal for the scalability and manufacturability of our device. Efficient testing, automated processes, and a focus on ethical labor practices are key to ensuring a smooth large-scale production process.

4.2.4.4 Recycling and End-of-Life Considerations

In the manufacturing of our multipurpose device, addressing recycling and end-of-life considerations is vital for environmental sustainability and ethical practices. To reduce electronic waste and minimize our environmental impact, we prioritize designing the device for easy disassembly, utilizing recyclable and eco-friendly materials, collaborating with e-waste recycling programs, implementing product take-back initiatives, and ensuring compliance with environmental certifications like RoHS. Additionally, consumer education on responsible disposal, adopting a circular economy approach, safeguarding data privacy during recycling, and maximizing component reusability contribute to a comprehensive recycling strategy. These initiatives align our device with the growing demand for eco-conscious and ethically manufactured technology solutions while promoting a sustainable product life cycle.

5.0 ChatGPT

The gaming industry has always involved the player interacting with the in-game artificial intelligence(AI). In the ever-evolving landscape of modern technology, the gaming industry has continually pushed the boundaries of innovation, delivering immersive experiences that captivate players worldwide. To further enhance this pivotal component, the integration of AI has emerged as a promising avenue for research and development. Here, we aim to explore the limitations, pros, and cons of utilizing AI-powered platforms like ChatGPT in the process of creating gaming controllers.

As AI technologies continue to evolve, they have found applications beyond their traditional domains. In research endeavors, AI platforms have become indispensable tools that assist in the generation of ideas, facilitate data analysis, and enhance the overall learning experience. However, it is essential to critically evaluate the potential benefits and drawbacks of incorporating AI, such as ChatGPT, into the development of our gaming controller to ensure informed decision-making.

This section will compare the limitations of using AI-powered platforms in gaming controller research, scrutinize the pros and cons, and provide real-world examples of how these platforms can either positively contribute to or potentially harm the learning experiences of individuals involved in controller development [**5.0a**]. By examining these aspects, we aim to shed light on the opportunities and challenges associated with AI's role in shaping the future of gaming controllers, ensuring that researchers and developers can harness its potential effectively while mitigating its risks.

5.1 Limitations

The integration of AI-powered platforms, such as ChatGPT, in research presents a host of limitations that must be considered to ensure a balanced and informed approach to development. These limitations can impact the effectiveness and reliability of AI-based solutions.

5.1.1 Creativity

AI platforms, like ChatGPT, operate based on pre-existing data patterns and algorithms. This means that they lack true creative thinking. In the field of gaming controller design, where innovation and creativity play a pivotal role, AI may struggle to generate groundbreaking concepts or designs that go beyond established norms. Human ingenuity remains irreplaceable in this aspect.

5.1.2 Bias

AI systems can inherit biases present in the data they are trained on, potentially introducing unfair or problematic elements into gaming controller designs. If not carefully monitored and corrected, these biases can lead to issues related to inclusivity, representation, and discrimination, which can negatively impact both the gaming industry and its audience.

5.1.3 Generalization

AI models excel in tasks for which they are trained but may struggle to generalize to new, unseen situations. In research, this limitation can restrict the applicability of AI to broader contexts and may require retraining models for different research objectives. ChatGPT's knowledge is based on information available up to January 2022. The AI model does not have access to events, developments, or information that has occurred after that date.

5.1.4 Lack of Understanding

AI systems like ChatGPT can generate human-like text, but they lack genuine comprehension of the information. They generate responses based on patterns in the data they were trained on, which can lead to incorrect or nonsensical answers.

5.1.5 Lack of Critical Thinking

AI models do not possess critical thinking skills, so they can't evaluate the quality or validity of the information they generate. Researchers often need to critically assess sources and information.

It is important to remember to use AI like ChatGPT as a tool, rather than a replacement for traditional research methods. It can be a valuable resource for generating ideas, summarizing information, or assisting with basic tasks, but it should be used in conjunction with human expertise to ensure the quality and validity of research outputs.

5.2 Pros and Cons

AI has become an integral part of our modern world, revolutionizing various industries and aspects of our daily lives. While AI brings numerous advantages, it also comes with its share of drawbacks. Striking a balance between harnessing the advantages of AI while mitigating its drawbacks is a crucial task for society and policymakers as AI continues to advance and proliferate.

5.2.1 Pros

There are many advantages of using AI to supplement your research. Here are some of them:

5.2.1.1 Data Analysis

AI can handle vast amounts of data, helping researchers organize, clean, and extract valuable insights from large datasets, which may be unmanageable through manual efforts alone. AI can uncover patterns, trends, and insights in large datasets that may be challenging for humans to identify.

5.2.1.2 Efficiency

AI can significantly speed up data analysis and processing mundane tasks, allowing researchers to focus on higher-level tasks. This leads to faster results and more efficient research processes. The use of AI is particularly beneficial in data-intensive research fields.

5.2.1.3 Automation

Repetitive and time-consuming tasks can be automated using AI. This allows researchers to allocate more of their time on critical aspects of their work. This includes data cleaning, image recognition, and even laboratory experiments, as demonstrated by robotic lab assistants.

5.2.1.4 Personalized Prompts

AI can assist in tailoring research to individual needs by recommending relevant papers, studies, or sources based on a researcher's preferences and interests.

5.2.1.5 Pattern Recognition

AI algorithms can identify complex patterns and trends within data, aiding researchers in giving insight and making more informed decisions.

5.2.2 Cons

While using AI for research offers numerous advantages, there are also several potential drawbacks and limitations to consider. Here are some of the cons of using AI for research:

5.2.2.1 Human Intuition

AI lacks the intuition, creativity, and contextual understanding that human researchers possess. Some aspects of research, such as problem formulation or hypothesis generation, may still require human input.

5.2.2.2 Bias and Ethics

AI algorithms can inherit biases present in the data they are trained on, potentially leading to biased research outcomes. Ethical concerns arise in using AI for research, particularly in areas like healthcare and social sciences.

5.2.2.3 Transparency

Many AI algorithms are complex and difficult to interpret, making it challenging to understand how they arrive at their conclusions. This lack of transparency can hinder the reproducibility and trustworthiness of research results.

5.2.2.4 Dependence on AI

Over reliance on AI may lead to a reduction in critical thinking and creativity among researchers, potentially limiting the diversity of approaches to a problem.

5.2.2.5 Errors

AI systems are not always correct. AI systems can sometimes produce unexpected or unintended consequences, and these can be difficult to anticipate. Researchers must be proactive in monitoring and addressing such issues.

5.3 Enhancing Learning Experience

AI platforms, including systems like ChatGPT, have the potential to elevate user experiences through personalized information delivery, adaptive learning tools, and data-driven insights. It is imperative to acknowledge that while these platforms offer numerous advantages, their integration into our Senior Design course can also present potential challenges. This section will cover several examples that illustrate this concern and how it relates to the design of our multi-purpose device controller.

5.3.1 Personalized Recommendations

AI-powered platforms can analyze your performance, learning preferences, and past interactions with course materials to offer personalized recommendations. This can help identify relevant resources, assignments, or topics that require additional attention, tailoring the learning experience to specific needs.

5.3.1.1 Adaptive Learning Paths

AI can use intelligent insights to create personalized learning paths. For example, if the AI notices that you consistently excel in one area but struggle in another, it can adapt the course content accordingly. It might suggest additional resources, exercises, or readings for the topics where you need more support, while allowing you to move ahead in areas where you're already proficient. This tailoring ensures that your learning is both efficient and effective.

5.3.1.2 Recommendation Algorithms

Recommendation algorithms used in AI platforms are similar to those employed by platforms like Netflix and Amazon. They consider not only your individual performance and preferences but also aggregate data from other individuals with similar profiles. This collective information helps in making more informed and context-aware recommendations. Additionally, these algorithms continuously improve as they gather more data about your learning behaviors.

5.3.1.3 Continuous Feedback Loop

AI-driven recommendations don't stop at suggesting resources. They provide a continuous feedback loop for improvement. If you follow the recommendations, the AI platform can monitor your progress and assess whether your understanding or performance has improved. If not, it can adjust the recommendations or provide alternative resources. This iterative process ensures that you're continually supported in your learning journey.

5.3.1.4 Diverse Learning Resources

AI platforms can suggest a wide range of learning resources, such as textbooks, online courses, videos, articles, and interactive simulations. These recommendations encourage the user to explore different learning modalities, ensuring that a person is engaged with materials in ways that suit his/her learning style and preferences.

5.3.1.4 Time Management

Personalized recommendations can also help manage time more efficiently. AI can suggest a study schedule, allowing an individual to allocate time to the most critical topics and assignments, ensuring that any particular person is able to stay on track throughout the course.

5.3.2 Predictive Analytics and Resource Allocation

AI platforms can use predictive analytics to estimate the time and resources required for different project tasks. This helps project teams allocate their time and resources more effectively, ensuring that they meet project milestones and deadlines. By analyzing historical data and the progress of the project, the AI platform can provide recommendations on adjusting workloads, setting realistic expectations, and optimizing project planning.

5.3.2.1 Project Planning

Platforms such as ChaptGPT can incorporate a holistic view of a project by considering a wide range of factors, including task dependencies, historical data, resource availability, and individual team member capabilities. This comprehensive analysis allows for a more accurate estimation of the time and resources needed for each project task.

5.3.2.2 Resource Optimization

Through predictive analytics, AI platforms can suggest the most efficient allocation of resources. They can analyze the skill sets of team members, their workloads, and their availability to ensure that tasks are assigned to the right individuals. This optimization minimizes the risk of resource overallocation or underutilization, leading to smoother project execution.

5.3.2.3 Milestone and Deadline Management

AI can aid in setting realistic project milestones and deadlines. By examining historical project data and assessing the progress of the current project, the platform can make recommendations to ensure that milestones are achievable. It can also alert project managers to potential delays or bottlenecks, allowing for proactive adjustments to the project schedule.

5.3.2.4 Data-Driven Decision-Making

AI platforms can empower project managers and team members to make data-informed decisions. Rather than relying solely on intuition or anecdotal evidence, project decisions can be based on concrete data and insights. This data-driven approach can create a culture of continuous improvement and efficiency within the Senior Design course.

5.3.2.5 Adaptability

Predictive analytics is not a one-time process but an ongoing one. As the project progresses, AI platforms can adapt their resource allocation recommendations based on real-time data. This adaptability can allow project teams to respond to changing circumstances and ensure that they stay on track.

5.3.2.6 Optimized Collaboration

AI-driven resource allocation can improve collaboration within project teams. When resource assignments are transparent and aligned with individual strengths and workloads, team members are more likely to work better and efficiently. For our team project we used ChatGPT to formulate a game plan for our distribution of workload which gave us plenty of ideas we can implement for our project.

5.3.3 AI for Design Process

AI platforms played a pivotal role in catalyzing the commencement of our Senior Design project, focused on developing a multi-purpose device controller. ChatGPT and similar platforms provide invaluable support by aiding us in the creation of our initial design layout. By leveraging the power of AI, we were able to access creative and pragmatic design ideas. These ideas were not just simple concepts but also encompassed realistic and robust solutions that laid a strong foundation for our project. This initial phase of design layout became a crucial stepping stone for our team, offering us a clear and well-informed vision of how to embark on this complex endeavor.

While AI platforms greatly assisted us in generating ideas and concepts, we made a conscious and thoughtful decision not to overly rely on them. Instead, we aimed to strike a balance that would preserve the core essence of the learning experience within our Senior Design project. Recognizing that the technical and essential aspects of the project, such as programming and hands-on implementation, were central to our educational goals, we chose to retain full control over these processes. This approach ensured that the intellectual and practical aspects of our project remained in the hands of the designers – our team. By reserving these critical tasks for ourselves, we ensured that we gained in-depth knowledge and hands-on experience, in alignment with the primary objectives of the Senior Design course. In essence, AI platforms acted as valuable catalysts for our project's inception, guiding us toward a well-defined design layout while preserving the integrity of the learning process by refraining from automating the technical execution.

6.0 PCB

Our team for the project discussed in previous sections that the PCB is going to be the center of our work, the gaming controller is going to power up mainly from the Li-ion battery that our team chose for the controller. The Lithium-ion power bank will provide us with 5V and 2000 mAh, this is going to power MCU, LED, and MCU peripherals like the Bluetooth - WIFI chip.

In active mode the most power consumption the ESP32 is going to be drawing up from 400-500 mAh, assuming we are using all peripherals including Bluetooth and WIFI chip, and in the mode when the player is actively playing any game the ESP32 will be consuming around 160mAh.

Our LEDs that we choose for the microcontroller will be drawing the same voltage the power bank provides which is 5V.

The connection between our ESP32 microcontroller and lithium-ion power bank will require voltage regulation, as per data sheet instruction for ESP32-Wroom requirements the microcontroller voltage supply is 3.3 volts. This voltage regulator that we are using is going to be a step-down voltage regulator from 5v to 3.3 volts which is suitable for the MCU. Figure XX shows the design we are going to make for our PCB gaming controller and highlights the most important aspects regarding the power flow.

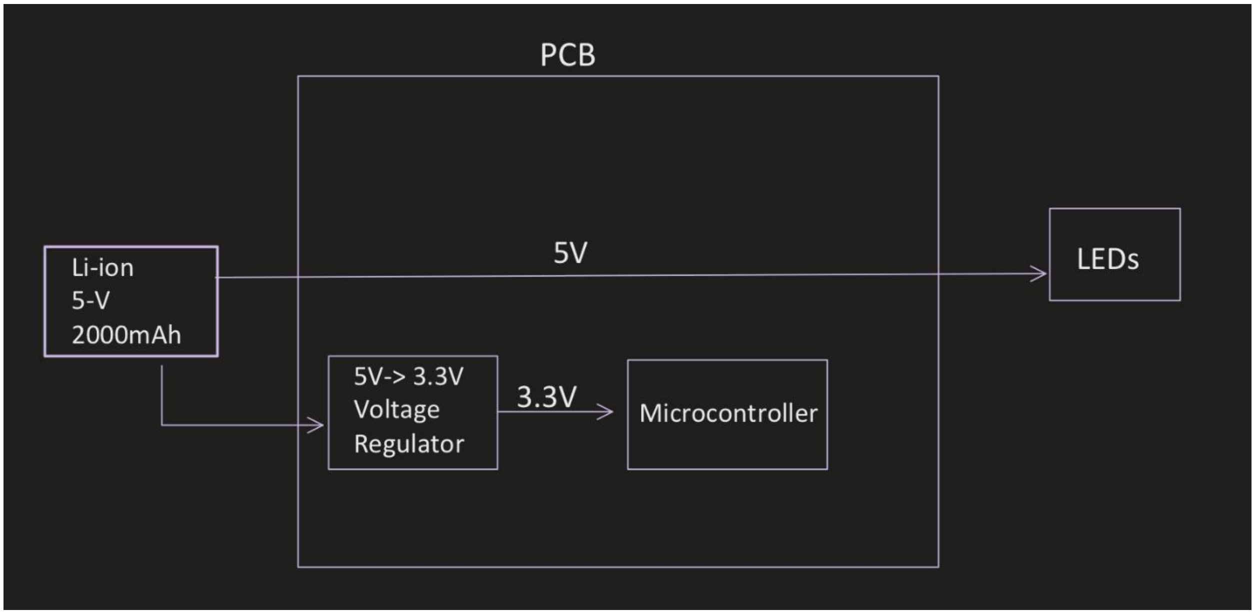

Figure 12 – Power Flow Diagram

6.1 Schematic Design

6.1.1 Voltage regulator (5V to 3V3)

For the next step, our job is to convert our design into a circuit. We have to design the voltage regulator. We have chosen Autodesk Eagle for our Schematic project design and also the final design.

In the process of designing the voltage regulator we have followed a few steps. First, we used Texas Instruments' WEBENCH power designer, where we had to enter the input voltage required and the output voltage and current required for our voltage regulator.

WEBENCH produces so many circuits that can work for the requirements we highlighted, so for the next step we have to choose which design will work best for our project. Our team decided that above 90% efficiency will be satisfying our needs for the voltage regulator. We also considered the Footprint of the part that we are choosing, although we are designing a gaming controller and size of the PCB will affect how our gaming controller looks like. Small parts can be hard to solder in the second part of our project. So, we choose TPSM863257RDX, with a 21.5mm square to be the key element of our voltage regulator.

In the process of choosing the design we also looked at BOM (Bill of Material) costs, and availability of the parts in the market. The design shown in figure xx will be inputting 5v and outputting 3.3v and 500mAh for our ESP32 to use.

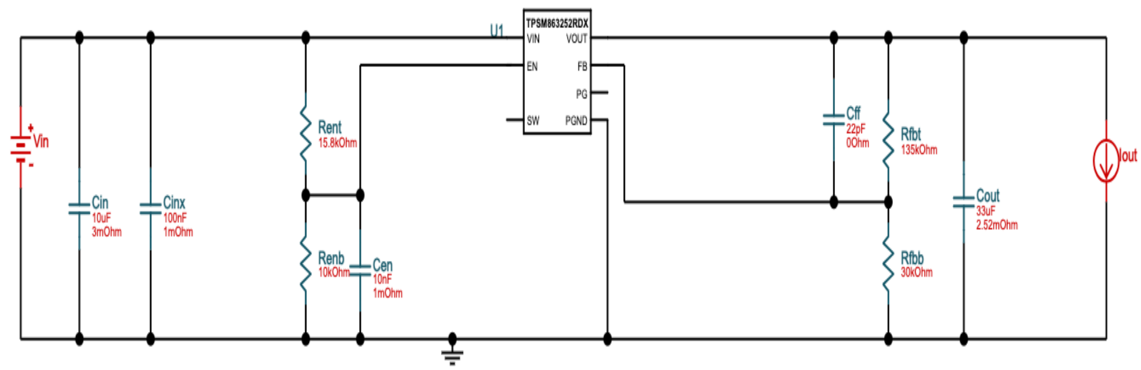

Figure 13 – 5V to 3.3V regulator Schematic

6.1.2 USB_C Connector

As our team mentioned above, we are going to use a usb_c connector to charge our gaming controller in general. The connector's job is to draw power from an external source and deliver it to our power bank, then the power bank is going to dispatch the power for the rest of the components. In the graph below we have the schematic design for the USB_C connector with the labels used to connect it to other circuits.

We followed the same process when generating the schematic design for our 5V to 3V3 voltage regulator. We looked up the part Digi-Key, the result matches hundreds of possible connectors. The next step we identified which connector would be good for our gaming controller. After choosing the part USB4105-GF-A, we looked at the data sheet for the component. The component we needed had to be in a certain criteria regarding the dimensions, and the circuit required to build the schematic. The dimension of the will help us later on to make the soldering process possible. The circuit required to build the schematic included two resistors, capacitor and a jumper which will need to add to the bill of material later after finishing the design.

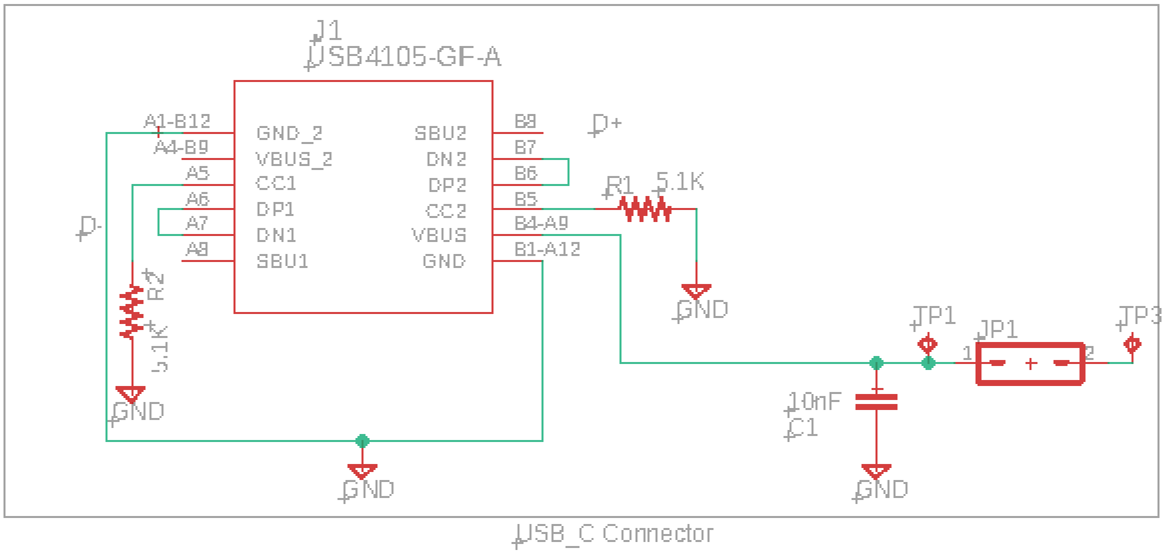

Figure 14 - USB_C Connector

6.1.3 ON/OFF switch

Before the power drawn from the wall to the usb_c connector gets to the power bank it has to bass through the ON/OFF switch. The switch will allow us to control the power flow to the power bank, when the switch is turned off the circuit will be completed and that means no power will flow from the usb_c connector to the power bank. This case will be when we have the power bank fully charged.

When the power bank is in low charge and the user connects the usb_c connector to a power source the power will flow to the power bank and charge, and the power bank will be dispatching power to the rest of the circuit.

This switch will provide safety to our PCB circuit and especially the power bank, where it guarantees there will be no constant current flowing to the power bank from the wall when the power bank is fully charged.

The component we chose for the ON/OFF switch is AP22804AM8-13, this will be part of the circuit along with three capacitors and one resistor.

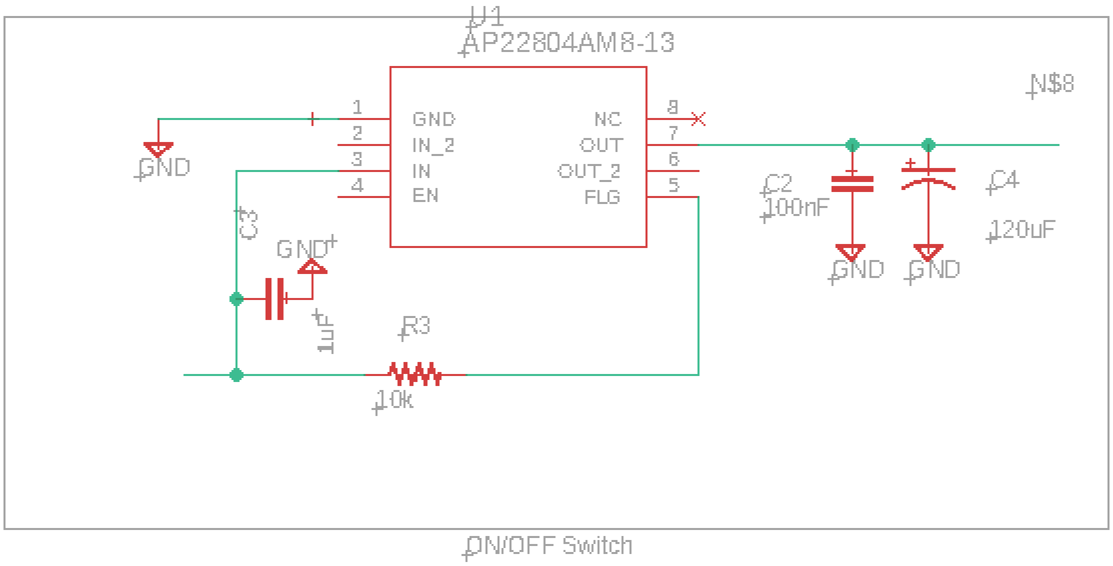

Figure 15 - ON/OFF Switch

6.1.4 Triggers

In the intricate design of our gaming controller, the quest for enhanced gaming precision leads us to a critical control mechanism—the TL3305AF260QG push button triggers. These triggers stand as a cornerstone in user interaction, shaping the responsiveness required for seamless execution of in-game actions, from precise aiming to executing complex special moves.

One standout feature of the TL3305AF260QG is its quick response time, a vital attribute for gaming controllers. As avid gamers know, split-second decisions can be the difference between victory and defeat. The TL3305AF260QG ensures instant actuation, translating player input into on-screen action with unparalleled speed. This quick response time becomes especially crucial for triggers like R1/L1 and L2/R2, where split-second reactions are often the key to success.

The TL3305AF260QG seamlessly integrates into the circuit. This precision integration is designed to optimize the overall system's functionality. The circuit's harmony ensures that each press of the trigger corresponds precisely to the intended action in the game, offering gamers a level of control that is both intuitive and responsive.

Gaming marathons demand durability, and the TL3305AF260QG rises to the challenge. Meticulously chosen for its robust construction, these push button triggers are built to withstand the rigors of extended gaming sessions. The tactile feedback from each press provides a reassuring feel, enhancing the overall gaming experience.

The TL3305AF260QG takes its place strategically, particularly as the chosen component for R1/L1 and L2/R2 buttons. In these critical positions, where precision, reliability, and quick response time are paramount, the TL3305AF260QG excels. Whether executing a rapid series of actions or delicately adjusting gameplay elements, these triggers deliver a level of control that enhances the gaming experience to new heights.

In essence, the TL3305AF260QG push button triggers contribute significantly to the gaming controller's success, combining quick response times, precise integration, and durability to elevate the gaming experience for enthusiasts seeking top-tier performance.

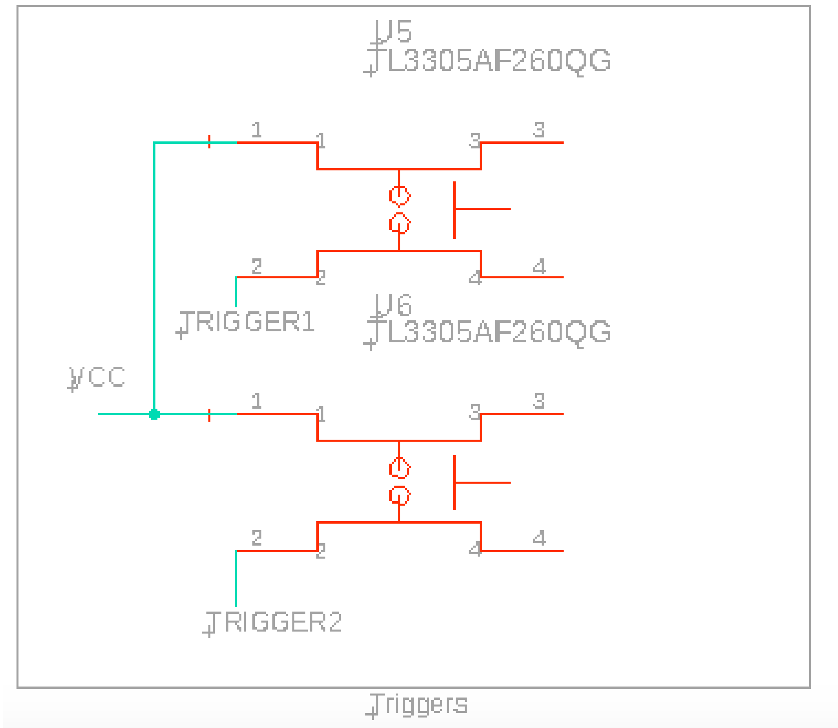

Figure 16 - Trigger

6.1.5 Action Buttons

Within the meticulous design of our gaming controller, achieving unparalleled precision involves a critical element—the B3F-1000 action buttons. These buttons serve as the linchpin in user interaction, defining the tactile responsiveness crucial for executing a myriad of in-game actions, from basic maneuvers to executing complex sequences.

A defining feature of the B3F-1000 action buttons is their swift responsiveness, a hallmark attribute sought after in gaming controllers. In the realm of gaming, where split-second decisions dictate success, these buttons ensure instant actuation. Swift and precise, the B3F-1000 buttons seamlessly translate player input into on-screen action, enhancing the overall gaming experience.

The B3F-1000 buttons effortlessly integrate into the circuit, marking a seamless marriage of precision and functionality. This integration is purposefully designed to optimize the controller's overall responsiveness. The harmonious relationship within the circuit guarantees that each press of the action buttons corresponds precisely to the intended in-game action, providing gamers with an intuitive and responsive control experience.

Crafted for the demands of gaming marathons, the B3F-1000 buttons stand resilient. Meticulously chosen for their robust construction, these action buttons are engineered to withstand the rigors of extended gaming sessions. The tactile feedback from each press not only endures but also enhances the overall gaming experience, providing a reassuring feel with every action.

The B3F-1000 buttons strategically take their place as the chosen component for the action buttons. Positioned with intent, these buttons deliver a trifecta of precision, reliability, and quick response time. Whether engaged in a rapid series of actions or delicately adjusting gameplay elements, the B3F-1000 action buttons empower gamers with a level of control that elevates the gaming experience.

In essence, the B3F-1000 action buttons play a pivotal role in the gaming controller's quest for excellence. Their swift responsiveness, seamless integration, and durable construction combine to set a new standard for gaming enthusiasts seeking top-tier performance. Each press of these buttons is an assurance of precision and control, contributing significantly to the gaming controller's success in delivering an exceptional gaming experience

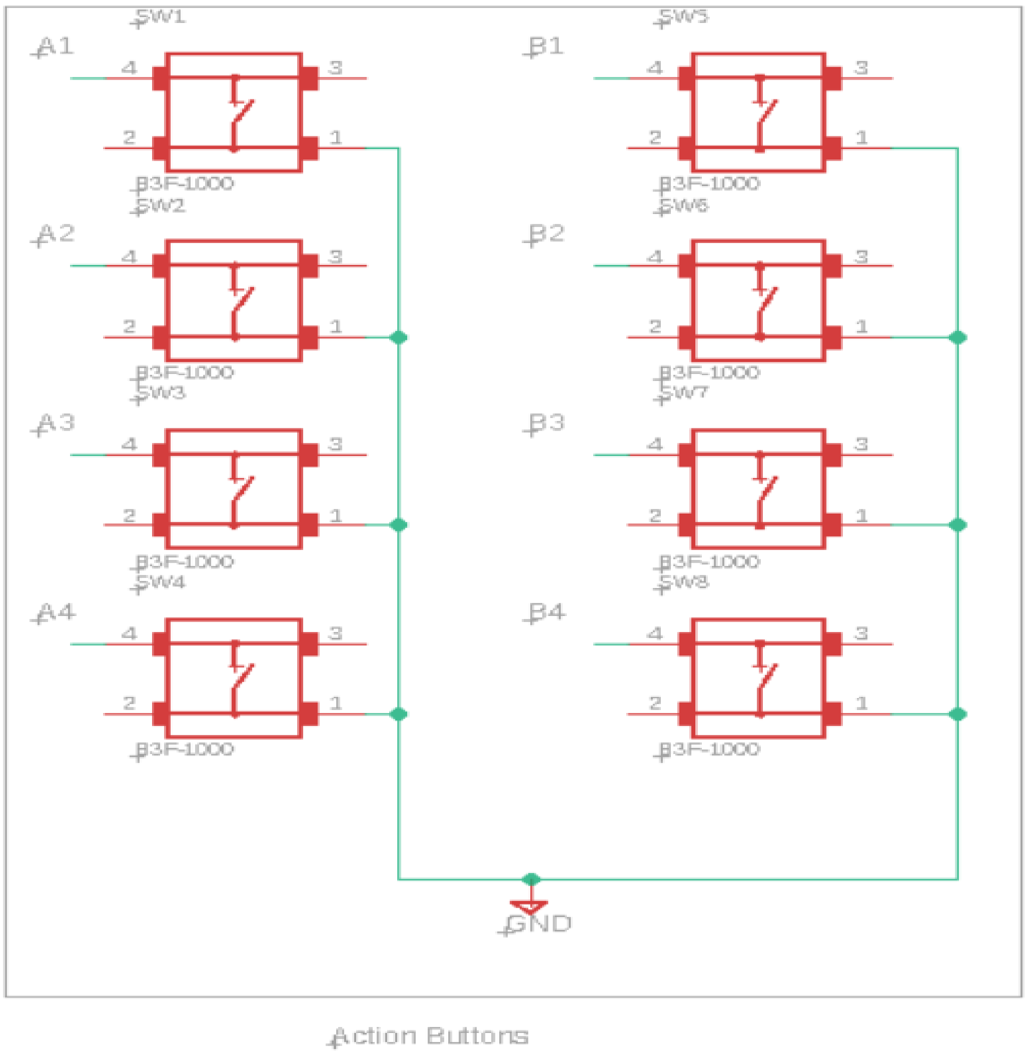

Figure 17 - Action Buttons

6.1.6 Analog Sticks

One of the most important components in our game controller's complex construction that allows for unparalleled accuracy is the COM-09032 analog sticks. These sticks are essential to user engagement because they provide the tactile response needed to perform accurate movements and moves in the game.

Using Hall Effect technology is one of the COM-09032 analog sticks' unique features. This novel method improves precision by offering dependable and accurate positional change sensing. The smooth integration of the Hall Effect sensors into the analog sticks guarantees accurate capture of even the smallest movements, resulting in a responsive and captivating gaming experience.

Like our action buttons, the COM-09032 analog sticks are known for their quick response times. These sticks guarantee immediate reaction in the gaming arena, where success is determined by split-second judgments. Because the analog sticks have Hall Effect sensors built into them, players can execute subtle movements with unmatched speed and accuracy.

The COM-09032 analog sticks fit into the circuit with ease, creating a well-balanced combination of accuracy and usefulness. The overall responsiveness of the controller is optimized by the careful design of this integration. The analog sticks in the circuit create a harmonious relationship when they are coupled to a common ground, ensuring accurate movement tracking and giving players a responsive and easy-to-use control system.

Designed to withstand the rigors of extended gaming sessions, the COM-09032 analog sticks are robust parts. They have a sturdy build that has been thoughtfully selected to survive extended gaming sessions. Every movement produces tactile input that not only lasts but also improves the overall gaming experience by giving each interaction a comforting sense.

The COM-09032 analog sticks, which are positioned strategically, are essential components for accurate analog control. These purposefully positioned sticks provide a blend of accuracy, dependability, and rapid reaction time. The analog sticks supply players with a degree of control that enhances the gaming experience, whether they are traversing complex environments or carrying out delicate in-game operations.

To put it simply, the COM-09032 analog sticks play a big part in the game controller's quest for performance. Their fast response time, smooth integration, robust build, and Hall Effect technology. These analog sticks provide control and accuracy with every movement, which is essential to providing an amazing gaming experience.

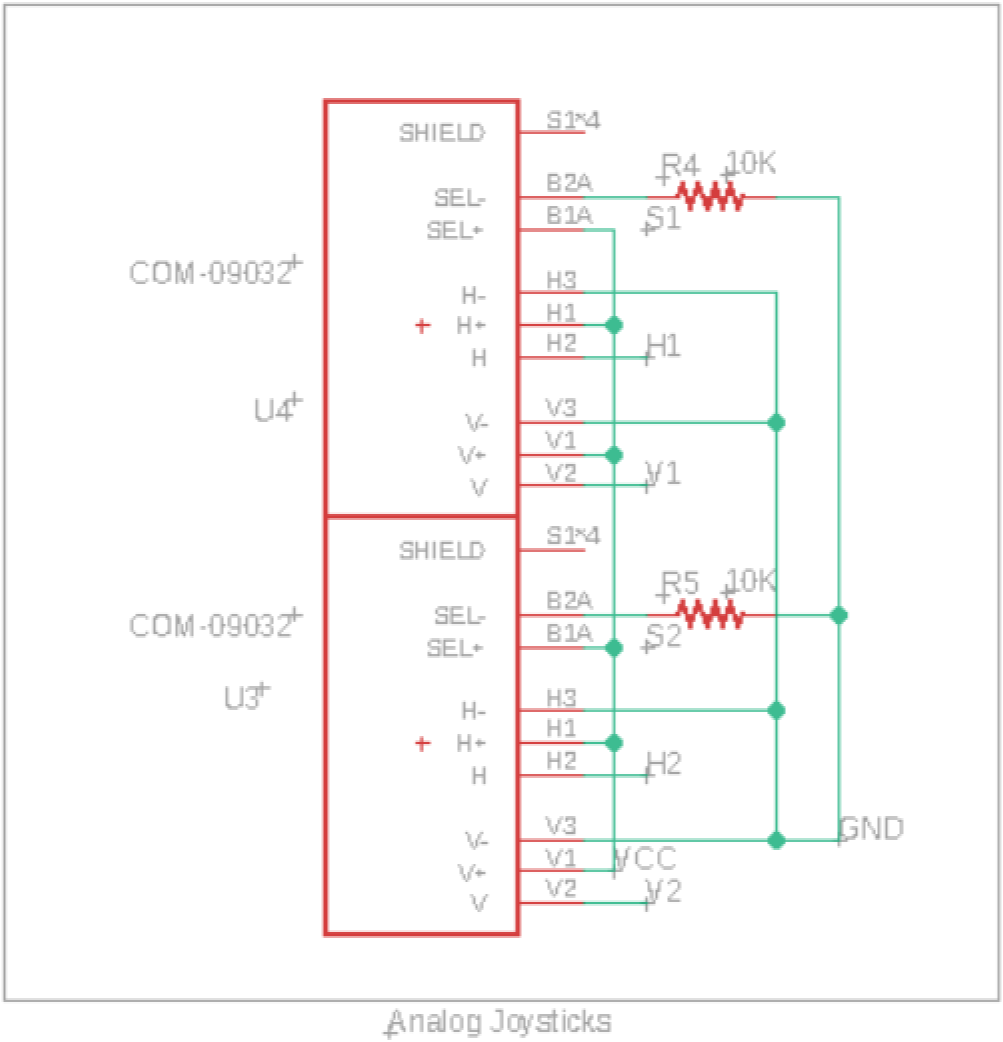

Figure 18 - Analog Sticks

6.1.7 Speaker

Within the complex architecture of our gaming controller, the SMS-1308MS-2-R speaker is a key element that provides unmatched accuracy. By delivering the rich and immersive audio experience required for contemporary gaming, this speaker serves as a sonic cornerstone and is vital for user engagement.

The SMS-1308MS-2-R speaker's ability to produce clear, sharp sound in spite of its small size is one of its best qualities. This small design is painstakingly made to fit perfectly inside the controller's boundaries, guaranteeing a rich sound experience without sacrificing the device's physical factor.

A remarkable technical achievement is the SMS-1308MS-2-R speaker's incorporation inside the gaming controller. With this connection, you can enjoy a more immersive and lively gaming experience overall. The speaker is cleverly positioned inside the controller to guarantee that every gaming detail—from explosions to footsteps—is accurately replicated, giving the player an even greater sense of immersion.

The robust audio produced by the SMS-1308MS-2-R speaker enhances the game experience. The speaker is a necessary component of an immersive gaming experience because of its capacity to accurately recreate soundscapes, which improves gameplay by offering important auditory clues.

The robust and long-lasting SMS-1308MS-2-R speaker is designed to withstand the rigors of extended gaming sessions. Because of its sturdy construction, which guarantees lifespan and dependability, it's a solid part for prolonged gaming sessions. Each sound that comes from the speaker has a clear resonance, which enhances the gaming experience.

Positioned carefully inside the controller, the SMS-1308MS-2-R speaker is essential to producing high-quality audio. Players will be fully submerged in the game's aural environment thanks to its small size and excellent audio reproduction. The subtle audio experience offered by the SMS-1308MS-2-R speaker includes both dialogue and background noise.

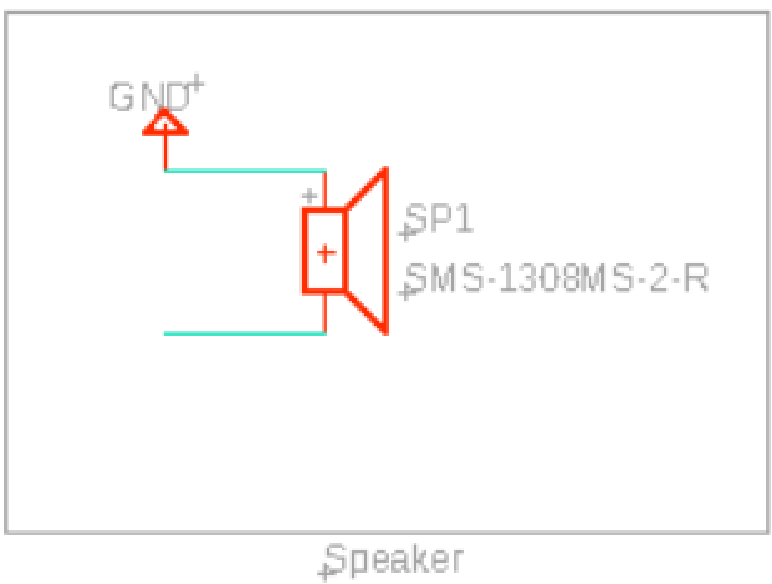

Figure 19 - Speakers

6.1.8 Microphone

In the intricate symphony of our gaming controller's design, a key player in achieving unparalleled accuracy is the CMA-4544PF microphone. This component stands as a crucial element for user engagement, offering impeccable sound capture capabilities that enhance the gaming experience. A standout feature of the CMA-4544PF microphone is its ability to deliver crystal-clear audio despite its compact size. This microphone is ingeniously crafted to seamlessly integrate within the controller, ensuring that users can communicate effectively with teammates or immerse themselves in the rich audio landscape of the game.

The CMA-4544PF microphone boasts an extended range, capturing audio nuances with exceptional clarity. Whether communicating in a multiplayer setting or recording in-game commentary, this microphone ensures that every vocal expression is faithfully reproduced, adding depth to the gaming experience. Designed to provide exceptional sound quality, the CMA-4544PF microphone elevates the auditory aspect of gameplay. The microphone's capability to capture clear and nuanced audio enhances communication, strategy discussions, and overall immersion in the gaming environment.

The compact design of the CMA-4544PF microphone allows it to seamlessly fit within the controller without compromising functionality. This integration is purposefully designed to optimize the overall gaming experience by delivering precise and reliable audio capture.

Crafted to withstand the rigors of extended gaming sessions, the CMA-4544PF microphone is a robust component. Its sturdy build ensures longevity and reliability, making it a dependable part for continuous and immersive gaming experiences. Every spoken word resonates with clarity, contributing to an enriched communication and gaming experience.

Strategically positioned within the controller, the CMA-4544PF microphone plays a vital role in facilitating communication. Its compact design, combined with superior sound capture capabilities, ensures that players can effectively convey messages, strategize with teammates, and fully engage in the social aspects of gaming. In essence, the CMA-4544PF microphone significantly contributes to the gaming controller's pursuit of performance. Its extended range, exceptional sound quality, compact design, and robust construction set a new standard for gaming enthusiasts seeking top-tier audio performance. Each spoken word is an assurance of precision and clarity, making the CMA-4544PF microphone an integral component in providing an exceptional gaming experience.

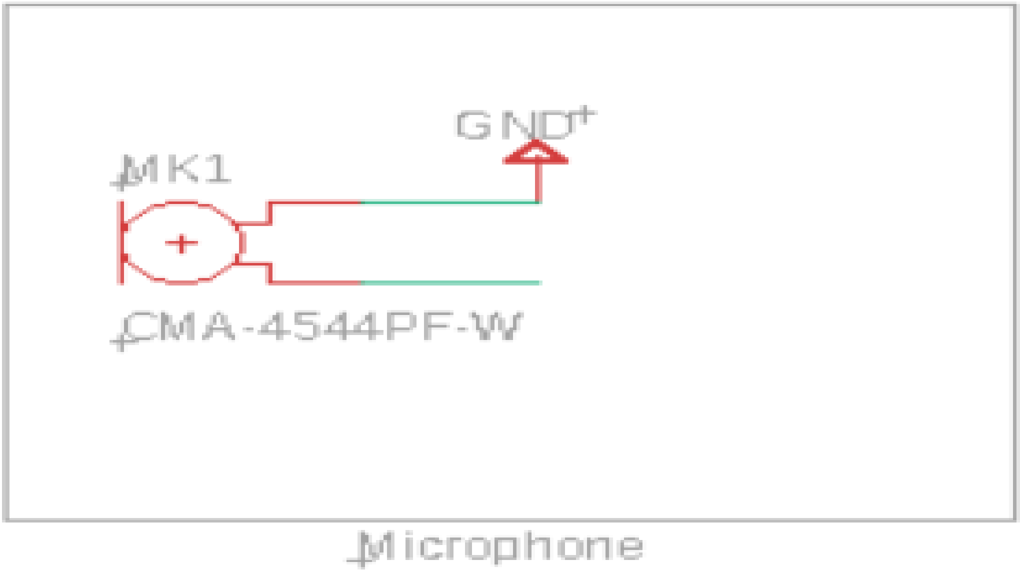

Figure 20 - Microphone

6.1.9 Switch Buttons

The switch buttons play a pivotal role in our configuration, providing a versatile means to establish connections to essential pins, namely IO0 and EN, on the MCU. When these connections are deemed unnecessary, the circuit intelligently closes, ensuring both pins are securely linked to ground. Within the intricate framework of our gaming controller circuit, the IO0 pin emerges as a linchpin. When ungrounded, it seamlessly transitions into a standard general-purpose input/output, offering multifunctionality. Notably, it doubles as a debugging pin for action groups, enhancing the versatility of our system. Additionally, IO0 assumes a critical responsibility in our PCB, serving as the conduit for external MCU resets. The strategic integration of a switch dedicated to the IO0 pin facilitates seamless and controlled resets of the MCU, augmenting the overall efficiency of our design.

Shifting focus to the switch associated with the EN pin, it operates in two distinctive modes, adding an extra layer of flexibility to our setup. In the first mode, when the enable is set to High, the ESP32 operates in its standard, full-capacity mode. During this state, it diligently executes the programmed code, ensuring the seamless operation of all designated peripherals. This mode is integral to the normal functioning of our system. Contrarily, the second mode comes into play when the enable pin is set to low. In this state, the ESP32 undergoes a reset or is effectively disabled. This feature proves invaluable in scenarios requiring a temporary halt in operations. Upon releasing the pin and restoring it to a HIGH state, the ESP32 undergoes a swift and automatic restart, initiating its functions afresh. This dynamic dual-mode operation, facilitated by the EN switch, underscores the adaptability and resilience of our ESP32-based system.

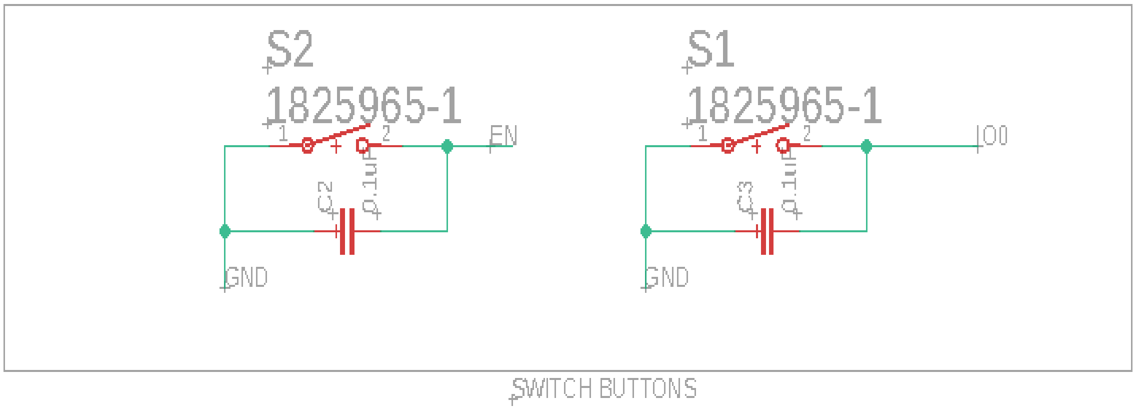

Figure 21 - Switch Buttons

6.1.10 Programming Circuit

A crucial component of the flawless integration and programming of the ESP32 microcontroller for gaming applications is the ESP32 Gaming Controller Programming Circuit. I provide an in-depth review of the fundamental components, connections, and practical information into the programming circuit.

Pinouts dedicated to gaming controllers are carefully incorporated within the programming header. This personalization accelerates the programming procedure while adapting to the specific requirements of the peripherals on the gaming controller.

During programming, a precise and controlled reset is ensured by a specialized reset and flash circuit. This is combined with a flash memory interface that works well to allow for quick and dependable firmware updates for the game controller.

Not only does a dedicated programming switch enable mode selection, it also starts programming. Users can swap between multiple firmware configurations with this special function, which makes the gaming controller more versatile for a wider range of gameplay conditions.

Mode Selection: By using the programming switch, users can customize the controller's functionality to fit their preferred gaming styles. Initiation and Reset: Turning on the programming switch starts the programming mode and causes the ESP32 to undergo a controlled reset. By doing this action, a reliable firmware update platform is ensured.

Firmware updates & Data Transfer: Effective data transfer is made possible by the programming header, which enables a smooth interface with external programming tools. Firmware upgrades are applied quickly, improving the game controller's functionality, and taking care of any new features or performance improvements.

Mode Switching and Completion: Users can utilize the programming switch to switch between gaming modes when the firmware upgrade is finished. Playing games without interruption is ensured by the reset and flash circuit, which provide a seamless changeover.

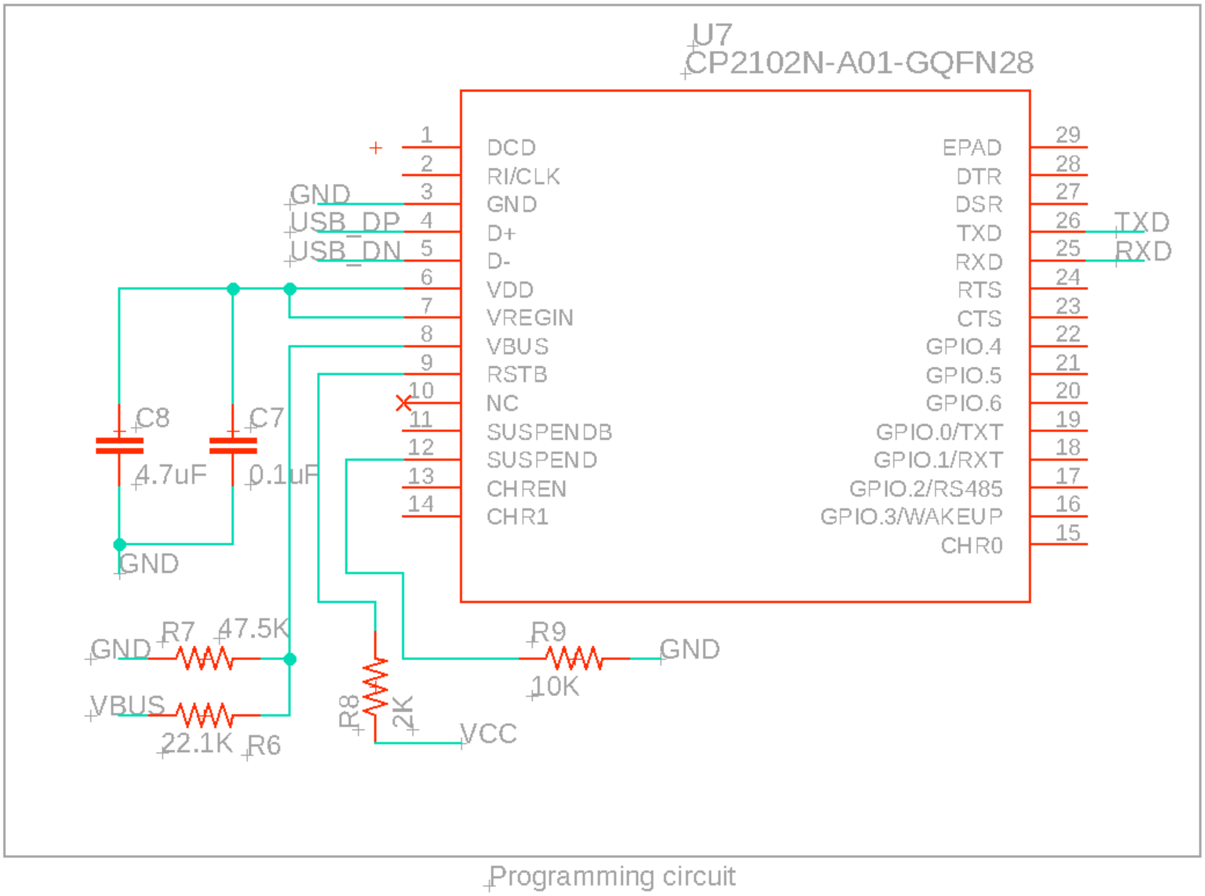

Figure 22 - Programming Circuit

6.1.11 Microcontroller Unit

In the orchestration of our gaming controller's intricate circuit, the ESP32 WROOM-32E takes center stage as the brain, orchestrating the seamless interaction of various components. This MCU serves as the nerve center, providing the computational power and connectivity required for the controller's sophisticated functionalities.

The IO pins of the ESP32 WROOM-32E act as a versatile interface, connecting and coordinating various elements within the gaming controller. These pins play a pivotal role in establishing connections to essential components, such as the triggers, buttons, speaker, and microphone. The strategic allocation of IO pins ensures efficient communication and interaction between different elements, contributing to the overall responsiveness and functionality of the gaming controller.

The ESP32 WROOM-32E's IO pins intricately connect to the push button triggers, ensuring swift and precise communication. These pins facilitate the quick response times and seamless integration of the TL3305AF260QG push button triggers, providing users with an unparalleled level of control during gaming sessions. Additionally, the IO pins orchestrate the connection with the B3F-1000 action buttons, enhancing the gaming experience through precision and reliability.

The ESP32 WROOM-32E's IO pins extend their influence to the auditory realm, connecting with the CMA-4544PF microphone and SMS-1308MS-2-R speaker. This connection ensures that the microphone captures clear audio signals, while the speaker faithfully reproduces immersive soundscapes. The IO pins contribute to the synchronization of these components, creating a harmonious auditory experience for the gamer.

Within the ESP32 WROOM-32E's comprehensive IO pin configuration, strategic emphasis is placed on pins associated with system resets. The IO0 pin, when intelligently grounded through a dedicated switch, transitions seamlessly between a general-purpose input/output and a debugging pin. This transition enhances the versatility of the system, adding functionality to the IO0 pin beyond its standard role. Moreover, the EN pin, controlled by a switch, operates in dual modes—enabling normal functioning when set to High and facilitating a swift restart when set to Low. This dynamic dual-mode operation ensures adaptability and resilience, allowing for controlled resets to enhance the overall efficiency of our ESP32-based gaming controller.

In essence, the ESP32 WROOM-32E MCU serves as the backbone of our gaming controller, orchestrating a symphony of interactions through its versatile IO pin configuration. This MCU's ability to seamlessly connect with and control various components, from triggers and buttons to the speaker and microphone, underscores its role as a central hub in delivering top-tier gaming performance. The strategic allocation of IO pins, facilitating dynamic transitions and resets, contributes to the adaptability and efficiency of our ESP32-based gaming system.

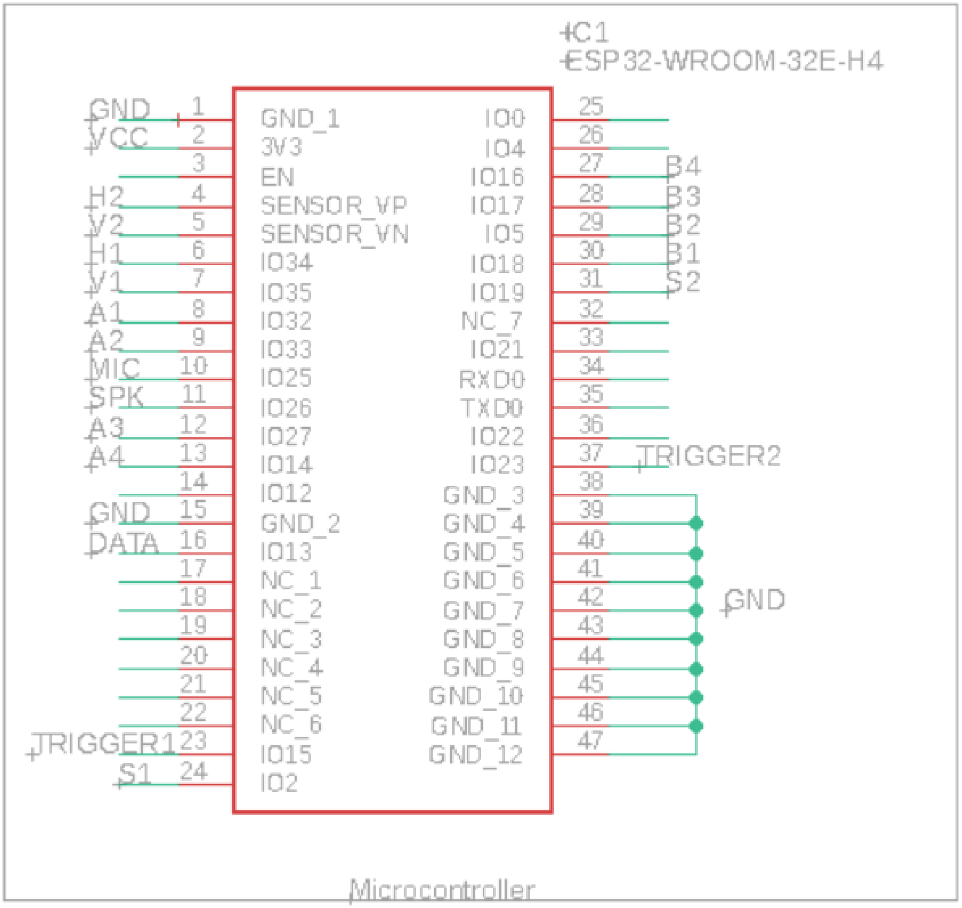

Figure 23 - MCU

Overall Schematic Design

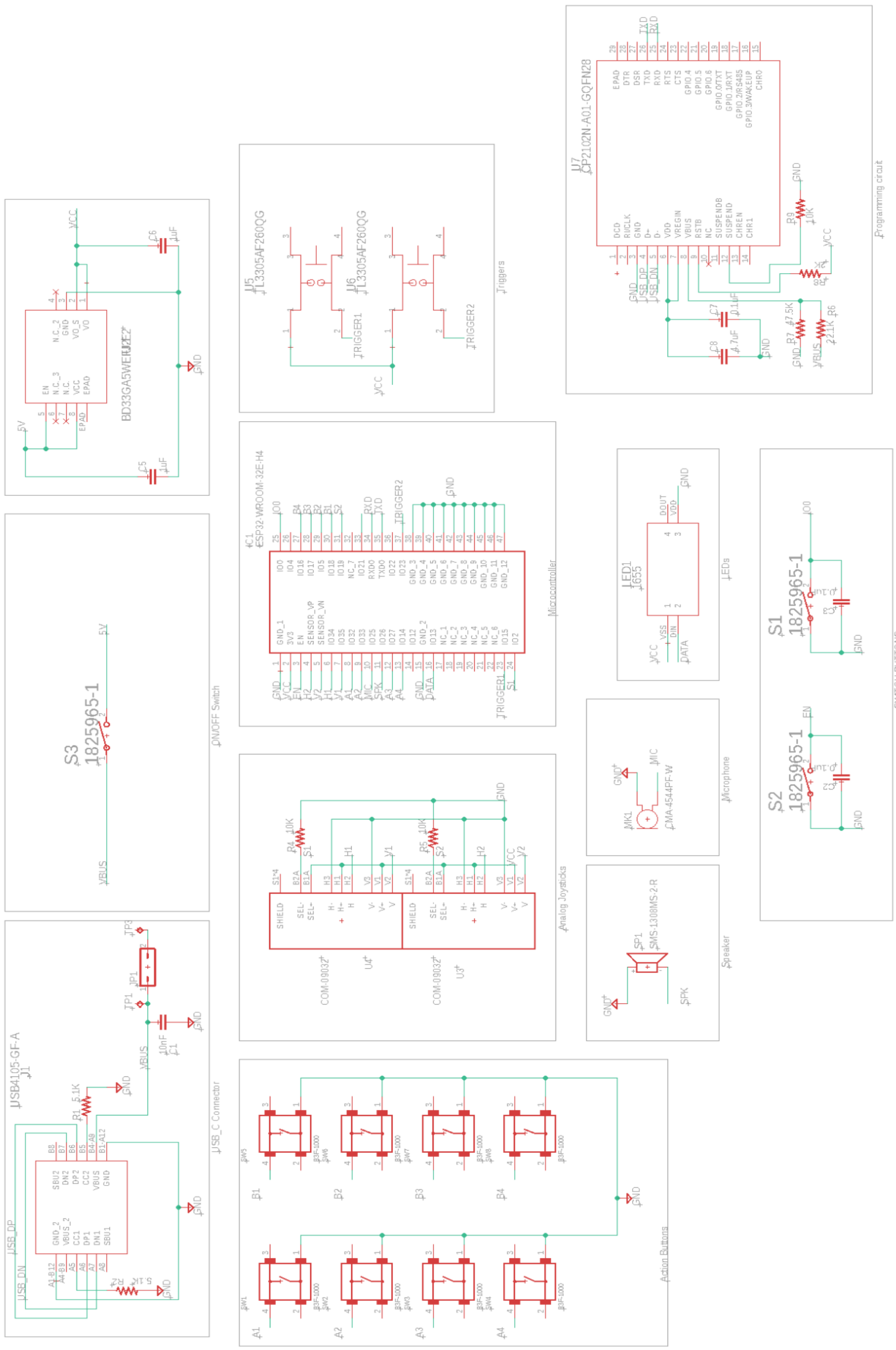

Figure 24 - Overall Schematic

7.0 Software Design

Crafting an innovative gaming experience involves more than just hardware. The soul of seamless connectivity and intuitive control resides within the software design. In this section, we delve into the intricate architecture and intelligent algorithms that power our gaming controller. Here, the ESP32 serves as the brains, harnessing its innate versatility to form the foundation upon which our controller's capabilities are built. Supported by the ingenuity of Bluetooth technology will create an ecosystem where devices seamlessly communicate and synchronize in perfect harmony. Moreover, it's the integration of carefully chosen libraries that amplifies this symphony, empowering our software framework to orchestrate a fluid interplay between the controller and multiple devices, transforming mere gaming sessions into immersive experiences that transcend the ordinary.

7.1 Software Purpose

What does the software of our controller do? The main functions of our software will include:

- **Connection Establishment -** Through a selection of ESP32 libraries, our controller seamlessly links up with gaming consoles. With BLE it ensures the console recognizes our controller as a familiar input device.

- **Console Switching -** Once a connection has been established, the user will have to manually switch the controller's configurations to its respective device. A stretch goal we have is to have the controller automatically detect what type of console it is connected to and switch to its respective control scheme.

- **LED Behavior -** We implemented LEDs to indicate battery life, whether the controller is on or not, and which controller scheme the controller is in.

- **Input Behavior -** The controller's inputs will be processed by the ESP32 and communicate to the console.

With these four essential functions seamlessly embedded within our controller's software, users will be able to connect and start gaming.

7.2 Prerequisites

In the implementation process of integrating these functions into our controller system, several crucial preparatory steps must be undertaken. Firstly, it's essential to understand the specifications of the board itself and its breakout board, especially regarding pin configurations. It's pivotal to ensure code compatibility by assessing existing code or libraries and making necessary modifications for seamless integration. Conducting thorough tests to verify proper communication between the ESP32 and the breakout board, and documenting the setup, connections, and troubleshooting steps, is essential for future reference and replication.

7.2.1 ESP32-WROOM-32

Bluetooth on the ESP32 is managed through its integrated Bluetooth module, allowing wireless communication with other Bluetooth-enabled devices. To start coding for Bluetooth functionality on the ESP32, you'll typically use a programming environment like the Arduino IDE or Espressif IDF. Firstly, you'll need to initialize the Bluetooth stack and choose between Bluetooth Classic or Bluetooth Low Energy modes based on your project requirements.

When using the Arduino IDE, the ArduinoBLE library simplifies the process of integrating BLE functionality. We'd begin by including the library and initializing the Bluetooth service. This involves defining services and characteristics that represent the data we will want to exchange between devices. Then we will set up the ESP32 to advertise itself as a peripheral or scan for and connect to other Bluetooth devices as a central.

Here are a list of libraries that will come in handy and/or use for establishing a connection:

Library	Purpose
ArduinoBLE	Simplifies Bluetooth Low Energy (BLE) setup on ESP32, facilitates creation of BLE peripherals/centrals, manages BLE services & characteristics.
ESP32 BLE Arduino	Extends ESP32's BLE capabilities within Arduino, offers additional functionalities, examples, and tools for handling BLE connections and data exchange.
ESP-IDF Bluetooth API	Comprehensive APIs for Bluetooth Classic and BLE on ESP32, provides fine-grained control over Bluetooth functionalities within the IoT Development Framework.
NimBLE (NimBLE-Arduino)	Lightweight BLE stack for ESP32 within the IDF, designed for resource efficiency, provides BLE features with minimal memory footprint, and supports Arduino-style integration.
BluetoothSerial	Enables ESP32 to function as a Bluetooth Serial Port Profile (SPP) device, facilitating serial communication over Bluetooth,

	allowing interaction with other devices.

Table 15 -ESP Libraries

7.2.2 Breakout Board

We decided to use a breakout board to aid us in the process of prototyping. Breakout boards offer several advantages, especially when prototyping with components like the ESP32. These boards simplify the utilization of individual components by essentially 'breaking out' the pins onto a separate, often smaller, circuit board. This design provides a labeled interface for easy connectivity, streamlining the process of linking these components to microcontrollers without the need for detailed pinout references.

One significant benefit emphasized is space efficiency. Breakout boards typically employ smaller integrated circuits, which occupy less space compared to their larger counterparts commonly used with breadboards. This efficiency becomes vital, particularly in projects where space optimization is crucial or when dealing with intricate circuit designs.

The AITRIP ESP32 Breakout Board presents a practical solution for prototyping with the ESP32 microcontroller, offering a streamlined approach to GPIO pin expansion. Its design allows the expansion of one GPIO pin into two, effectively doubling the available pins for connections. This feature proves invaluable in development requiring extensive pin utilization.

Moreover, this breakout board aligns with safety, reliability, and scalability, making it a dependable tool for various applications. Its ability to maximize pin utility allows for the reuse of all available pins on the ESP32, optimizing connectivity without compromising safety or stability during prototyping.

7.2.3 Selecting Programming Language for Device

To determine the right programming language for our Multi-Purpose Device Controller, the focus was on developing the software in an environment that facilitates communication between the gaming controller and the target device, often a gaming console, PC, or mobile device. This involves handling inputs from the various buttons, triggers, joysticks, and other components of the controller. The programming language and Integrated Development Environment (IDE) choice depend on the specific requirements and the compatibility of the target platform.

For Python, developers may use IDEs like PyCharm or Visual Studio Code to write scripts for controller functionality. However, Python is not as common for low-level gaming controller programming, although it is versatile and can be employed for tasks like handling input events or creating custom controller functionalities.

In Java, Eclipse or IntelliJ IDEA might be used for programming gaming controllers. Java is suitable for developing software components that interface with controllers, providing a higher-level programming environment.

If the controller is designed for a gaming console or platform that supports C#, Visual Studio becomes a prominent choice. C# is commonly used for game development, and Visual Studio provides a robust environment for coding the software that interprets controller inputs.

JavaScript, often used for web-based gaming, can be programmed using IDEs like Visual Studio Code or WebStorm to handle controller inputs in the context of browser-based gaming.

In essence, we wanted to choose a programming language that was desirable for our device which will include writing the logic and drivers that interpret button presses, joystick movements, and other inputs, translating them into actions within a game or other applications.

7.2.4 Arduino IDE / C++

We decided that we'd be using the Arduino IDE with C++ to program our project. The language and environment offers a straightforward and accessible platform for developers. Its simplicity is a key draw, providing an environment that's welcoming to beginners and experienced programmers alike. This IDE abstracts complexities, offering an easy-to-understand interface that accelerates the development process.

What makes Arduino particularly powerful is its vast community support, providing a wealth of tutorials, forums, and libraries. These libraries are a significant advantage, streamlining the integration of sensors, devices, and other components into projects. Additionally, the platform's agility in rapid prototyping is unparalleled, allowing swift creation and testing of concepts. While Arduino simplifies programming concepts, it's based on C++, providing a stepping stone for developers to advance towards lower-level functionalities as their expertise grows. Its cross-platform compatibility ensures developers can work seamlessly across different operating systems, further enhancing its appeal as a versatile and accessible programming environment for the ESP32.
In order to enable programming functionality for the ESP32, the installation of a driver tailored for the USB-to-UART chip is crucial. This choice hinges upon the specific chip integrated into the board, which could either be the CH340 or CP210x chip. These chips, manufactured by Silicon Labs, differ in their technical specifications and functionalities. In our setup, utilizing the CP210x chip, we need to install the corresponding driver provided by Silicon Labs. These drivers, meticulously crafted by Silicon Labs, serve as a bridge, facilitating seamless communication between the ESP32 board and the programming interface. Their role is pivotal in ensuring a stable and efficient connection, empowering developers to harness the full potential of the ESP32 in their programming endeavors.

Another essential step involves modifying the board manager URLs within the Arduino IDE. These URLs serve as pathways or links that direct the IDE to access and retrieve necessary board-related information and configurations. In our case, we've added an additional board manager URL specifically designated for the ESP32. This modification allows the Arduino IDE to access and integrate the required tools, specifications, and settings tailored for the ESP32 board. It ensures that the IDE recognizes and supports the ESP32, enabling developers to effectively write, compile, and upload code to this particular hardware platform directly from the

Arduino development environment. This enhancement streamlines the development process, offering seamless integration and compatibility for programming the ESP32 within the familiar Arduino IDE environment.

The final step involves installing the ESP32 within the board manager of the Arduino IDE. The board manager acts as a repository or library of hardware specifications and configurations specifically tailored for various development boards. By incorporating the ESP32 into the board manager, provided by Espressif Systems in this case, we enable the Arduino IDE to access and manage all the necessary tools, drivers, and settings essential for programming the ESP32. This integration within the board manager ensures that the IDE comprehends the nuances of the ESP32 board, allowing developers to seamlessly write, compile, and upload code directly targeted for this hardware platform. Essentially, integrating the ESP32 into the board manager facilitates a streamlined development process by providing a comprehensive and optimized environment for working with the ESP32 in the Arduino IDE.

Here's a recap of the additional downloads:

Component	Purpose
USB-to-UART Driver (CP210x by Silicon Labs)	Facilitates communication between the computer and the ESP32 board via the USB interface. Enables programming by establishing a stable connection.
Additional Board Manager URL (ESP32)	Allows the Arduino IDE to access and retrieve specific tools, settings, and configurations tailored for the ESP32, enabling seamless integration and programming.
ESP32 Board Package (Espressif Systems)	Incorporates the ESP32 hardware specifications, drivers, and tools within the Arduino IDE's board manager, facilitating coding, compilation, and uploading tasks.

Table 16 - ESP Prerequisites

7.3 Implementing the Functions

After the prerequisites have been met and the environment has been set up we can begin prototyping the key functions of our controller.

7.3.1 Connection Establishment

The initial phase in facilitating Bluetooth communication involves the establishment of a connection between two or more devices. This pivotal step serves as the foundation for the exchange of data, enabling seamless interaction and information transfer across Bluetooth-enabled entities.

Establishing a Bluetooth connection entails a series of intricate procedures orchestrated within the Bluetooth protocol stack. This intricate stack comprises multiple layers, each meticulously handling aspects of the connection initiation, authentication, and establishment phases. It commences with the activation of the Bluetooth modules on the respective devices, triggering a process of device discovery wherein the devices actively seek and identify each other within the vicinity.

The Bluetooth protocol stack operates as a structured hierarchy of layers governing the intricacies of communication between Bluetooth-enabled devices. At its core lies the Physical Layer (PHY), responsible for the transmission and reception of radio signals defining wireless communication. Above it, the Link Layer (LL) manages connections, packet handling, and error correction. Further up, the Host Controller Interface (HCI) acts as the bridge between the hardware and software, facilitating communication between the Bluetooth controller and the host device. Sitting atop the stack are layers like the Logical Link Control and Adaptation Protocol (L2CAP), Security Manager (SM), and Attribute Protocol (ATT) coupled with the Generic Attribute Profile (GATT), responsible for segmenting data, providing security measures, and structuring data exchange between devices.

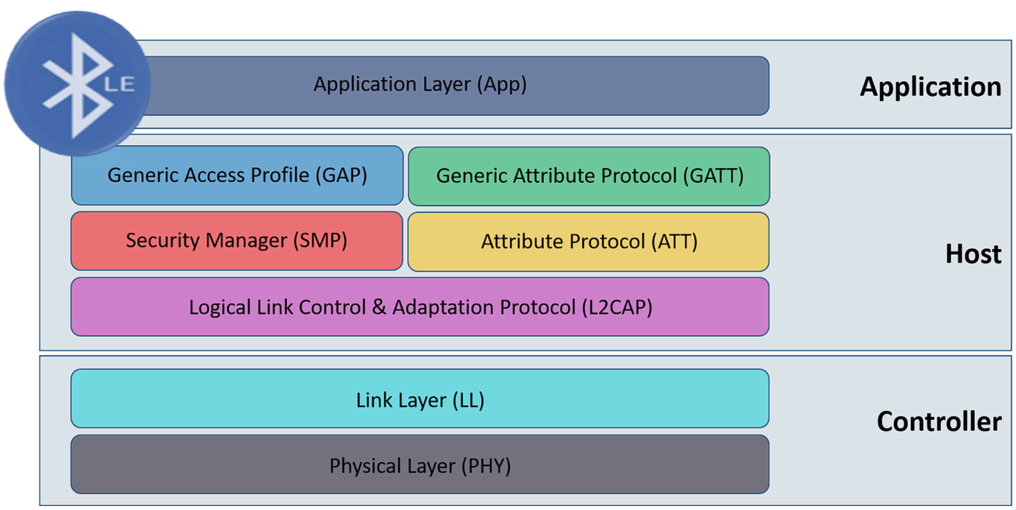

Figure 25 - Bluetooth Protocol Stack [7.3.1a]

When the ESP32 initiates a connection with other devices via Bluetooth, it navigates through these layers. Beginning with initialization, the ESP32 configures its Bluetooth module, setting up the PHY and Link Layer functionalities. Through the HCI layer, it issues commands to the Bluetooth controller, managing device discovery, connection establishment, and data transmission. The L2CAP layer handles packet formatting, ensuring efficient data exchange. If security measures are necessary, the Security Manager enables authentication and encryption. Finally, using the GATT layer, the ESP32 organizes data into services, characteristics, and descriptors, facilitating interaction with the functionalities offered by the connected device. This comprehensive understanding of the Bluetooth protocol stack will aid us in the prototyping of finding a connection between other devices.

After understanding the Bluetooth protocol stack and how the ESP32 initiates connections with other devices we can create a function that initializes discovery and communication between another device via Bluetooth. Here is a software flowchart with a general idea of how it will work.

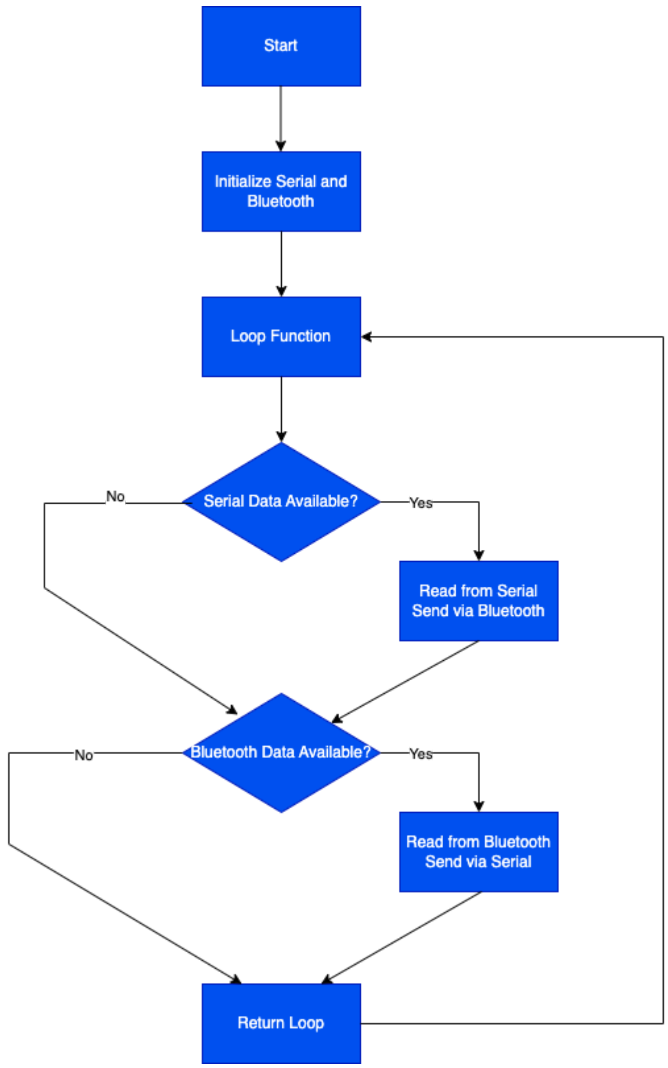

Figure 26 - Bluetooth Initialization

This flowchart starts with the setup phase, initializing both Serial and Bluetooth, then enters the loop. In the loop, it continually checks for data availability on Serial and BluetoothSerial. If data is available on Serial, it is read and sent via Bluetooth. Conversely, if data is available on BluetoothSerial, it is read and sent back via Serial. This process continues in a loop.

7.3.2 Console Switching

Another key function of our gamepad will be the ability to switch devices that it is connected to.The gamepad's console-switching functionality introduces a user-friendly approach, leveraging a designated button combination during power-up to transition between different console profiles. This feature allows users to dictate the target console or device the gamepad connects to, simplifying the process with a familiar and intuitive action.

When the gamepad powers up, the user engages a predefined button combination, typically by holding down a specific set of buttons while turning on the device. This action serves as a trigger, signaling the gamepad's firmware to activate the console-switching mode, prompting the gamepad to enter a specialized configuration state.

Within this configuration state, the gamepad recognizes the console profile designated by the user through the button combination. It then initiates a tailored connection sequence, identifying and establishing a connection with the corresponding console or device associated with that profile.

This approach ensures an efficient transition between different console profiles without the need for complex setups or manual reconfiguration. Moreover, the use of distinct button combinations for each console profile streamlines the process, offering users a straightforward and easily memorable method to align the gamepad with their desired gaming platform.

To enhance user experience and provide clear feedback, the gamepad may employ visual cues such as LED indicators to confirm successful profile selection, ensuring users are promptly informed of the active console profile upon power-up.

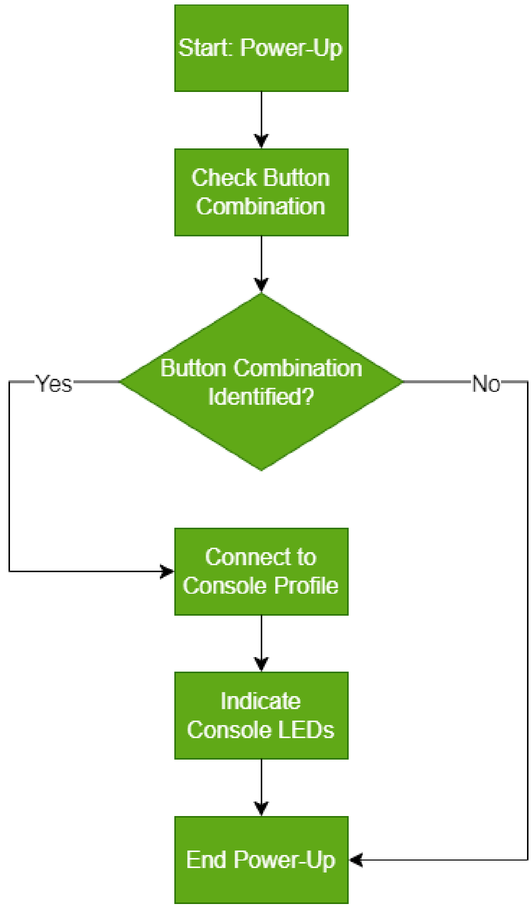

Figure 27 - Console Switching

The flowchart focuses on detecting a specific button combination during the power-up phase to enable console switching. The Check Button Combination function oversees this process, monitoring for the specific button combination pressed during power-up. Once detected, it proceeds to identify the associated console profile, initiating a connection attempt with that console using the Connect To Console Profile function. Feedback mechanisms are employed to confirm successful console switching via the Indicate Console LEDs function or to indicate connection failures. Should the button combination not be triggered during power-up, the gamepad follows its regular power-up sequence, ensuring normal operations without altering console profiles.

7.3.3 LED Behavior

We aim to optimize our LED indicators on the gamepad to convey essential information to users. These LEDs will serve a multi-functional purpose, indicating crucial details such as the remaining battery life, the active console profile currently in use, and whether the controller is charging or not. By utilizing distinct light patterns or colors, users will grasp vital information, ensuring they're informed about the battery status, the selected console profile, and whether the controller is currently undergoing a charging process. This intuitive LED system intends to enhance user experience by providing clear, at-a-glance information.

Initially, our design centered around utilizing a single LED for the entire controller, assuming it would suffice for conveying various information. However, upon closer consideration, we recognized the potential limitations this approach might pose. We realized that relying solely on a single LED would severely restrict the ways in which we could effectively communicate essential information to users. For example, we can't have the battery life indication and the console profile indication on at the same time because the controller shares one LED. This realization prompted us to reevaluate our strategy, acknowledging the importance of an expanded LED system that could offer diverse light patterns or multiple indicators.

We're faced with a pivotal decision: to either retain a single LED while incorporating distinct flashing patterns to signify both battery life and the charging status, or to opt for a dual LED setup. In the case of the latter, one LED would be dedicated to conveying battery-related information—such as remaining charge and charging status—while the other would serve the exclusive purpose of indicating the active console profile. These options present differing approaches: the single LED with varying patterns for comprehensive data signaling, or a dual LED system for more streamlined and dedicated information representation.

We've opted for a simplified LED configuration, utilizing a single indicator to convey multiple crucial details. Here's the breakdown: the LED will maintain a constant color to represent the active console profile (with specific colors pending finalization). To communicate the battery status, the LED will display a flashing pattern tailored to the battery's level. For instance, if the battery life is high, the LED will flash green every 5 seconds for half a second before returning to the console profile color. This cycle continues, with each specific color flash—green for high battery life, yellow for medium, and red for low—providing users with a clear understanding of the remaining charge. Moreover, during the charging process, the LED will briefly flash blue every 5 seconds for 0.5 seconds, promptly resuming its display of the console profile color. This refined LED system ensures users receive distinct and straightforward visual cues regarding both the console profile and the battery status without unnecessary complexity.

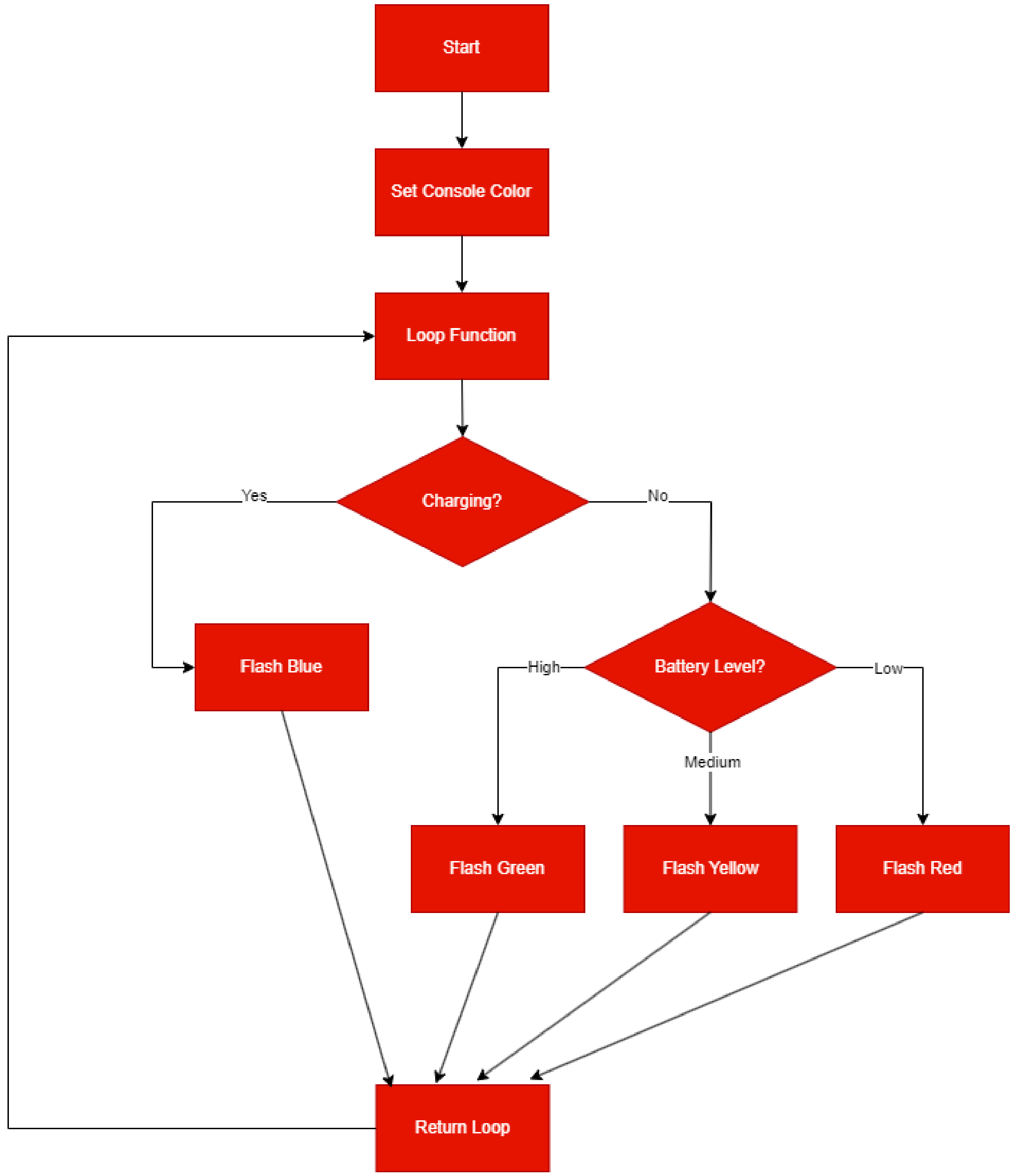

Figure 28 - LED Behavior

The software begins by initializing the LED to reflect the current console profile color. Following this initialization, a continuous loop starts, first checking whether the gamepad is currently charging. If it is, the LED flashes in blue, indicating the charging status. In the event the gamepad isn't charging, the loop moves on to assess the battery level. Depending on the battery level—high, medium, or low—the LED flashes with the corresponding color (green for high, yellow for medium, and red for low battery levels). Subsequently, the loop restarts, perpetually monitoring the charging status and battery level, ensuring the LED provides real-time feedback about the console profile and battery status.

7.3.4 Input Behavior

In our gamepad, we've incorporated various inputs, each serving a distinct function and contributing to the overall user experience. These inputs encompass a range of elements, including triggers, action buttons and analog sticks. Despite all these inputs being processed by the microcontroller (MCU), the resulting actions or outcomes they trigger are diverse and tailored to their specific functionalities. Each input element plays a unique role in the gamepad's functionality, contributing nuanced and varied responses.

7.3.4.1 Triggers and Action Buttons

In the gamepad's design, despite the physical distinctions between triggers and action buttons, they serve a similar purpose in terms of gameplay functionality. Both triggers and action buttons act as input mechanisms to initiate specific in-game actions or functions. Triggers, typically positioned at the rear of the gamepad, often control actions such as firing a weapon or accelerating in racing games. On the other hand, action buttons, often placed on the front face of the controller, serve similar gameplay functions, enabling actions like jumping or interacting with in-game objects.

The buttons on our controller are directly linked to the printed circuit board (PCB), which then connects to specific pins on the ESP32 microcontroller. Mapping functions to these buttons becomes a straightforward process of accessing and utilizing these designated pins within the ESP32. By calling upon these pins, we establish the mapping between the physical buttons and the corresponding actions or functionalities programmed into the system.

When programming the triggers and action buttons on a gamepad, the approach tends to be similar despite their physical differences. Both elements follow a similar programming paradigm focused on event handling and mapping specific actions to each input.

In configuring the gamepad's buttons, we've chosen to employ a polling technique to manage their input. This method revolves around the constant querying of button states at regular intervals rather than relying on interrupt-driven detection. The polling process involves systematically checking the status of each button in the gamepad across consecutive cycles, detecting any alterations in their state—whether pressed or released.

This technique brings forth several advantages. Firstly, it ensures predictability in the timing of input checks, enhancing the system's responsiveness by allowing precise control over when button states are monitored. Secondly, its simplicity in implementation simplifies the code structure, making it more understandable and easier to manage. Additionally, polling tends to be compatible across various platforms and microcontroller configurations, offering versatility in its application. Moreover, in scenarios with limited hardware resources, polling can be more resource-efficient compared to interrupt-driven methods as it doesn't rely on continuous interrupts for handling input changes.

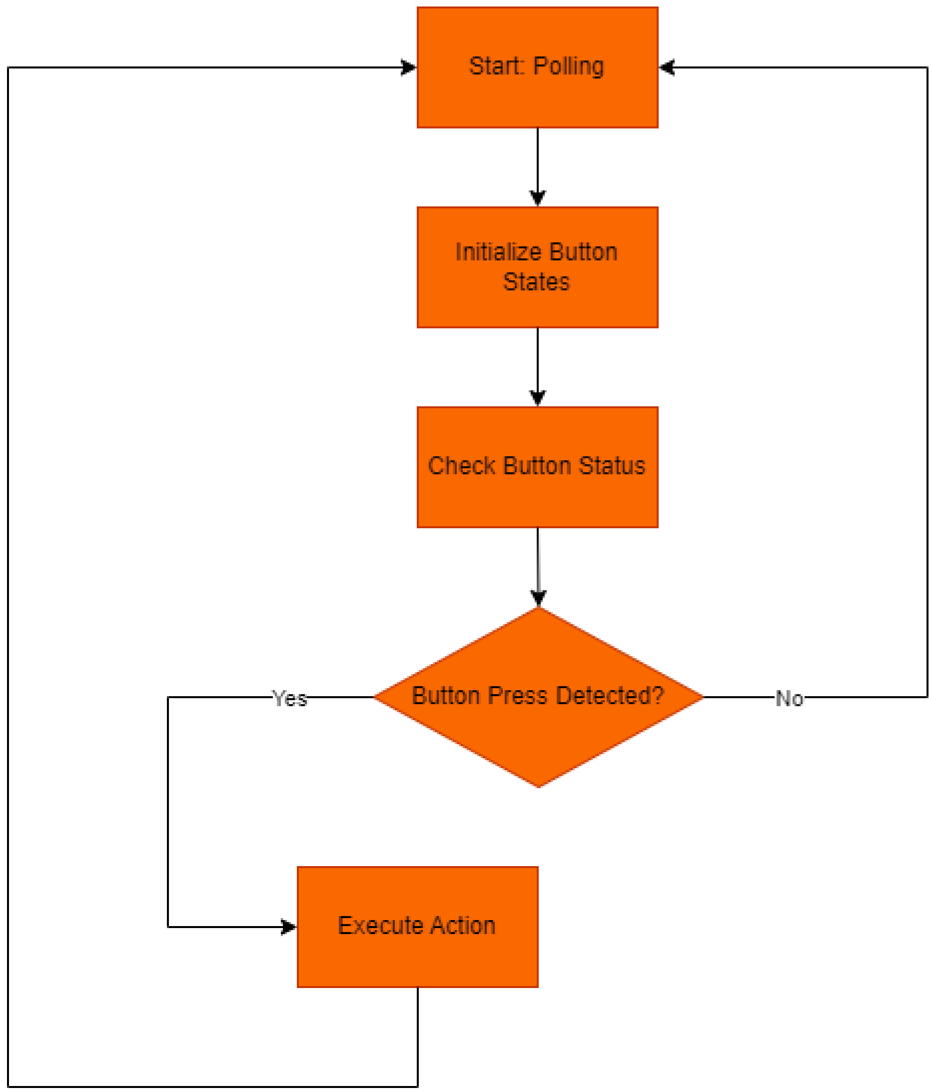

Figure 29 - Button Behavior

The flowchart begins by initializing the button states, priming them for ongoing monitoring. Continuously, it checks the status of each button, ready to detect any press. Upon detecting a button press, it triggers the associated action linked to that specific button. Conversely, if no button press is detected during the polling cycle, the flow proceeds without executing any actions. Ultimately, after each polling cycle, the flow concludes by updating the game's actions based on the detected button presses, ensuring a responsive and dynamic user experience in the gameplay.

7.3.4.2 Hall Effect Joysticks

Our controller's standout feature lies in its Hall effect joysticks—an advanced input mechanism that operates using the Hall effect principle. These joysticks employ a Hall effect sensor to detect the position of the joystick lever, translating magnetic field variations into precise voltage outputs. This design ensures accurate and durable control inputs, offering a contactless and reliable method for navigating games or controlling devices.

We may plan to integrate a dead zone configuration for our joysticks. It is an essential feature that defines a specific neutral region around the joystick's center position. This dead zone ensures that slight joystick movements near the center won't register as inputs, eliminating unwanted sensitivity in that area. It's crucial to implement a dead zone to prevent unintentional or erratic actions caused by minor joystick shifts or natural hand movements. By setting this neutral zone, users can experience smoother and more precise control, especially in scenarios where precise and deliberate movements are necessary, such as aiming in games or navigating interfaces. This feature enhances user experience by eliminating unintended actions, providing a more controlled and accurate input mechanism.

Programming a dead zone for a joystick involves detecting and disregarding small variations in the joystick's resting or neutral position to prevent unintended actions or jittery movements.

The general approach to implementing a dead zone is:

1. Read Joystick Values: Continuously read the joystick's X and Y axis values (often ranging from 0 to 1023 for analog joysticks) to determine its position.

2. Identify Dead Zone Range: Decide on a dead zone range around the center position of the joystick. For instance, a dead zone might extend, for example, ±50 units around the center value (e.g., if the center value is 512, the dead zone would span from 462 to 562).

3. Check Joystick Position: When reading the joystick values, check if the values fall within the dead zone range for both X and Y axes.

4. Ignore Values within Dead Zone: If the joystick values fall within the dead zone range for both axes, set the joystick's output to the neutral position (usually the center value) or disregard these values, preventing any action or movement.

5. Process Non-Dead Zone Values: If the joystick values fall outside the dead zone range, process these values as usual to control the intended actions or movements.

This logic helps filter out small joystick movements around the center position, ensuring a smoother and more controlled user experience by eliminating unintended actions within the dead zone range.

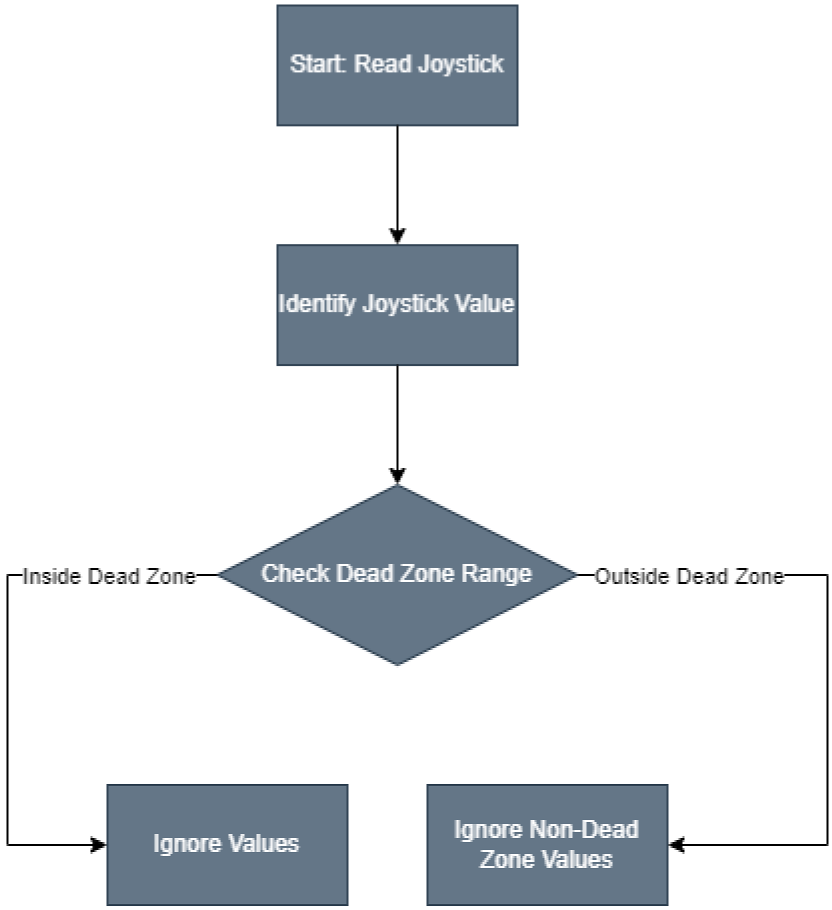

Figure 30 - Deadzone Configuration

The sequence commences by capturing the joystick's X and Y axis values, extracting the present joystick data. It then examines whether these values reside within the predefined dead zone surrounding the joystick's center position. Should the values exist inside this zone, indicating minor joystick adjustments near the center, the process either dismisses these values or sets them to neutral. However, if the values extend beyond the dead zone, signaling substantial joystick movements, the sequence proceeds to process these non-dead zone values for controlling actions or movements. Ultimately, the sequence finalizes its operations by managing actions based on non-dead zone joystick values. Adjustments or commands executed outside this neutral area remain unaffected by minute movements within the dead zone, ensuring a heightened level of precision and controlled user input.

Enclosing these steps within a loop structure allows for ongoing monitoring of the joystick input, ensuring that the dead zone logic is continuously applied to provide smooth and accurate control within the desired range of joystick movements.

7.4 Software Summary

Our software framework is designed around four pivotal functions to ensure seamless operation and user interaction. Firstly, the "Connection Establishment" function leverages specific ESP32 libraries to facilitate a smooth and reliable link between our controller and gaming consoles. Secondly, the "Console Switching" functionality empowers users to manually toggle between different controller configurations tailored to specific devices. Additionally, we aspire to achieve an advanced feature—automatic detection of the connected console type, allowing the controller to intelligently switch to the respective control scheme without user intervention.

Our third key function, "LED Behavior," introduces a visual cue system through LEDs integrated into the controller. These LEDs serve multiple purposes: indicating battery life, signaling the controller's power status, and displaying the active controller scheme, ensuring users are informed at a glance.

Lastly, the "Input Behavior" function orchestrates the processing of controller inputs by the ESP32, enabling communication with the console. Each input received from the controller undergoes meticulous processing, ensuring accurate transmission and interpretation by the connected console.

In preparation for implementing these functionalities, we've outlined the prerequisites: access to the ESP32 microcontroller and its associated libraries, utilization of a breakout board for hardware connections, and leveraging the Arduino IDE with its C++ programming environment.

By defining these core functions and prerequisites, we aim to create a sophisticated and user-centric controller system that enhances gaming experiences through seamless connectivity, intelligent operation, and clear visual feedback.

8.0 System Fabrication/Prototype Construction

8.1 PCB Layout Importance for a Gaming Controller

The PCB layout stands as a pivotal element in the intricate process of crafting our gaming controller. Its significance is deeply rooted in its ability to meticulously manage the communication of signals coursing in and out of the controller, thereby mitigating potential complications such as electromagnetic interference (EMI). This becomes particularly crucial as our gaming controller interacts with various electronic components and devices. The PCB layout acts as a guardian, diligently filtering out unwanted signals, ensuring that the controller functions seamlessly and without disruptions.

Furthermore, the PCB layout plays a critical role in guaranteeing the stability and reliability of power distribution within the gaming controller. Power delivery, whether sourced from the wall or a battery in a power bank, undergoes a complex journey through various pathways and components. The PCB layout serves as the orchestrator of this intricate dance, ensuring that each element is properly connected and that power flows efficiently and predictably. This meticulous management of power distribution not only enhances the controller's performance but also contributes to its overall longevity and reliability.

As we transition to the assembly phase, the impact of a well-designed PCB layout becomes even more pronounced. The ease with which components can be soldered onto the PCB and seamlessly integrated into the targeted platforms is directly influenced by the layout's effectiveness. A well-thought-out layout not only simplifies the physical assembly process but also streamlines the interconnection of components, reducing the likelihood of errors and enhancing overall efficiency.

Moreover,The advantages of a well-designed PCB layout extend beyond the assembly phase.In the realm of senior design, where precision and optimization are paramount, a proficiently executed layout facilitates low latency. This directly translates into improved response times for the gaming controller, a critical factor in user experience. The efficiency in managing heat dissipation, another attribute of a well-designed PCB layout, ensures that the controller operates at an optimal temperature, preventing overheating and potential performance issues.

In essence, the PCB layout is the silent conductor orchestrating the symphony of signals, power, and components within our gaming controller. Its meticulous design not only shields the device from potential disruptions but also streamlines the assembly process and enhances the overall performance, ensuring our gaming controller stands as a testament to precision and functionality in the realm of electronic design.

8.2 Why Eagle?

The Eagle environment, crafted by Autodesk, emerges as an invaluable asset in the intricate journey of designing projects, especially for students. Autodesk's creation has garnered widespread popularity in academic settings, offering a host of advantages tailored to the unique needs of student projects. One of the standout features is the provision of a free version explicitly designed for students. This not only eases the financial burden on students but also ensures accessibility to essential features crucial for project development, eliminating the need to invest in premium plans.

In the realm of component integration, Eagle shines brightly. The platform boasts an extensive array of online libraries housing a diverse range of components— a treasure trove for students embarking on projects like our gaming controller. These libraries not only simplify the process of finding and incorporating specific components but also contribute to the versatility of the design process. The availability of components in various shapes and forms empowers students to select the most fitting designs for their gaming controllers, fostering creativity and customization.

Beyond its standalone advantages, Eagle has established itself as a cornerstone in many educational institutions, particularly in engineering programs. The the program's inherent simplicity is a key factor in its widespread adoption. The user-friendly environment minimizes the learning curve, enabling students to swiftly familiarize themselves with the tools and functionalities. This ease of use becomes especially pivotal when compared to alternative programs, where students might encounter challenges navigating more complex interfaces.

For many students, including our own team, Eagle serves as a crucial introduction to the hardware development environment for electronics. Its straightforward interface and intuitive design facilitate a seamless transition into the world of PCB design, providing a foundational experience that lays the groundwork for more complex projects. The program's popularity among engineering students speaks to its effectiveness in bridging the gap between theoretical knowledge and practical application.

8.3 PCB Layout Process

The initial step in our project journey is to ensure the accuracy of the schematic diagram. This involves running an ERC (Error Rate Check) to identify and eliminate any errors. It's crucial to get everything right at this stage before moving on to the next. Once the schematic diagram is error-free and all components are in place, we're ready to transition to the PCB design page.This shift is initiated by hitting the Generate/Switch to Board button, marking a pivotal moment in the project where the virtual plan starts transforming into the physical layout of our gaming controller.

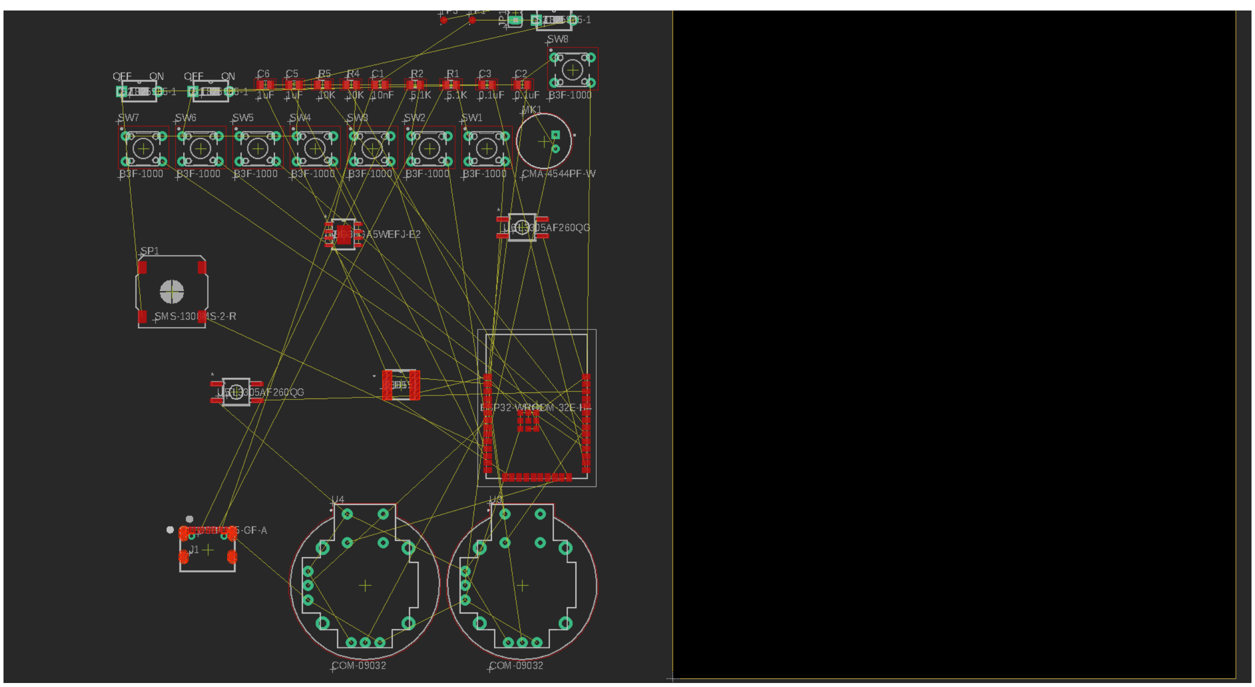

Figure 31 - PCB Layout

Step 2, after Switching to the board.

8.4 Components Placements

Designing an effective gaming controller hinges on strategic placement of its components, a crucial consideration for ensuring a seamless user experience. Among the most pivotal elements are the action buttons and joysticks, as their location significantly impacts user comfort during extended gameplay. The goal is to mitigate hand fatigue and provide a naturally relaxed grip for users, promoting an enjoyable gaming session.

Equally important is the placement of these components in a way that aligns with natural hand movements. This strategic positioning enhances intuitive control, allowing users to effortlessly interact with the gaming controller without the need to awkwardly stretch their fingers. This not only contributes to a more comfortable gaming experience but also ensures precision during gameplay, especially in scenarios where swift and accurate execution of actions is paramount.

While action buttons and joysticks take the spotlight, the location of other components, such as the MCU (Microcontroller Unit), plays a critical role in the overall functionality of the gaming controller. Placing the MCU at the heart of the board is akin to the central nervous system of the controller. This central location facilitates easy tracing and connection of all components to the MCU. By positioning it strategically, the MCU becomes a focal point, simplifying the intricate web of connections and ensuring efficient communication between various elements on the board.

In essence, the careful placement of components in our gaming controller design is not only about user comfort and gameplay precision but also extends to the organizational efficiency of the internal architecture. It's about creating an environment where the user can effortlessly engage with the controller, while the controller's internal components work harmoniously to deliver a responsive and enjoyable gaming experience.

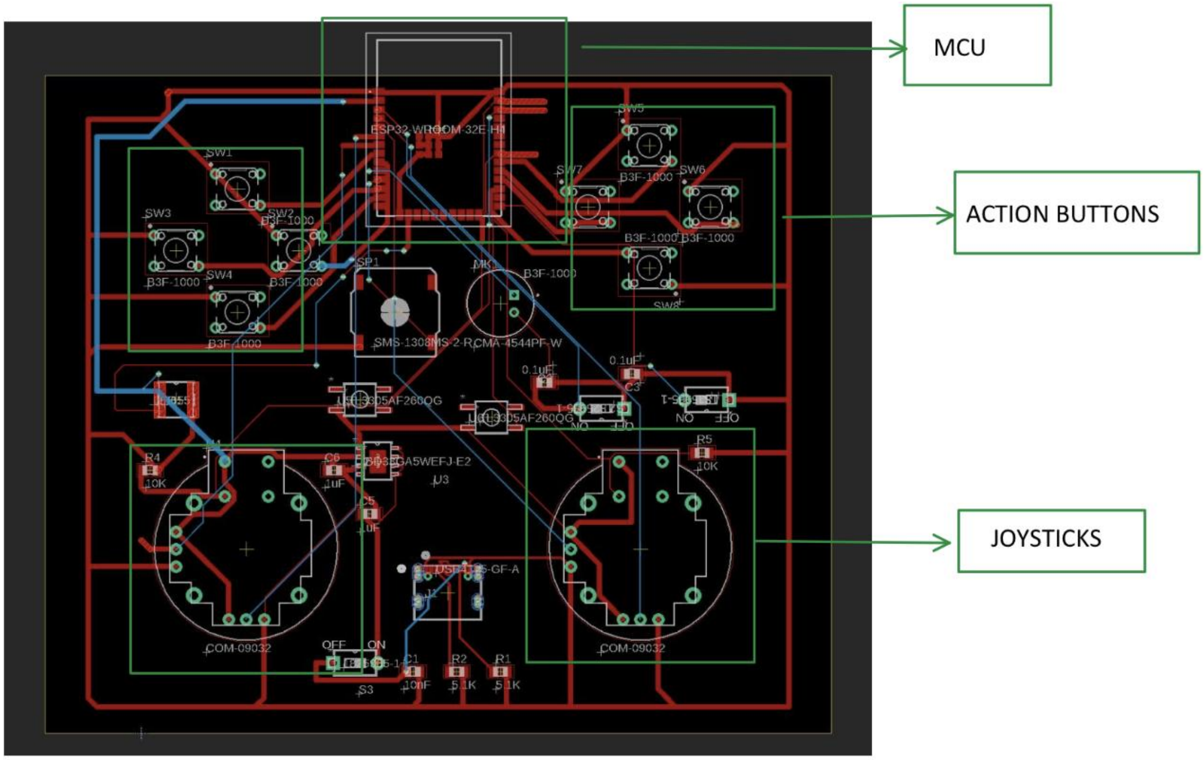

Figure 32 MCU, JOYSTICKS, and ACTION BUTTONS placement

8.5 Trace Routing

Trace routing is an important step in the design, where the power routes starting from the USB-C connector have to go through multiple components, for that we had to ensure that power routes are relatively larger than all other traces and our group decided to set it at a width of 30mil. Also ground traces are as important as power traces and we decided that it would be easier for the connection and design if we circle a ground trace around all the components.

The importance of large traces is crucial in PCB layout success for many reasons. First, the larger the width of the trace the less impedance it will have, and also the less resistance it will have. This will help to minimize the voltage drop throughout the trace route which will prevent any voltage drop from the start of the route to the end of it. Moreover, the dissipated heat in larger traces helps to ensure there will be heat efficiency, where high currents might pass through the route of power and ground, this can easily lead to performance issues and also damaging our PCB if we are not careful. Taking into consideration these elements will enhance our power delivery and signal integrity for a better design.

We used the autoroute tool to make the traces for other traces, like communication and signal traces, where the width of the signal trace is smaller than ground and power traces because it doesn't carry as much current and power in it. The traces that were made were either top of the

PCB or the bottom of the PCB. For the top we selected 1 TOP from the properties of the trace line, and for the bottom we selected 16 bottom from the same menu.

8.6 Vias Stitching and Grounding Planes

Vias stitching and grounding planes stand as indispensable techniques within the realm of PCB board design, collectively working towards the common goals of enhancing signal integrity, mitigating electromagnetic interference (EMI), and establishing a stable ground reference. In our meticulous pursuit of a gaming controller design, these practices become crucial cornerstones.

The concept of grounding planes involves the creation of expansive, continuous copper areas on both the top and bottom layers of the PCB. This dedicated ground space serves as a dependable reference point, offering a low-impedance pathway for the return of any current. The strategic placement and connection of these grounding planes around the PCB play a pivotal role in minimizing signal noise, providing an efficient return path for currents, and ultimately fortifying the reliability and performance of the entire circuit.

On the other hand, via stitching, a complementary technique, comes into play within the expansive spaces between components on our PCB. By weaving a network of vias in these strategic locations, we bolster the integrity of ground and power distribution. Much like grounding planes, via stitching is instrumental in creating low-resistance paths for the smooth return of currents. This technique becomes particularly pertinent in large PCB designs where the distribution of power and ground signals needs to be optimized.

Basically,both via stitching and grounding planes contribute synergistically to the overarching objective of a well-designed and high-performance PCB. The marriage of these techniques not only ensures a stable ground reference but also elevates the overall reliability and efficiency of the circuit, aligning seamlessly with the stringent requirements of advanced PCB board design.

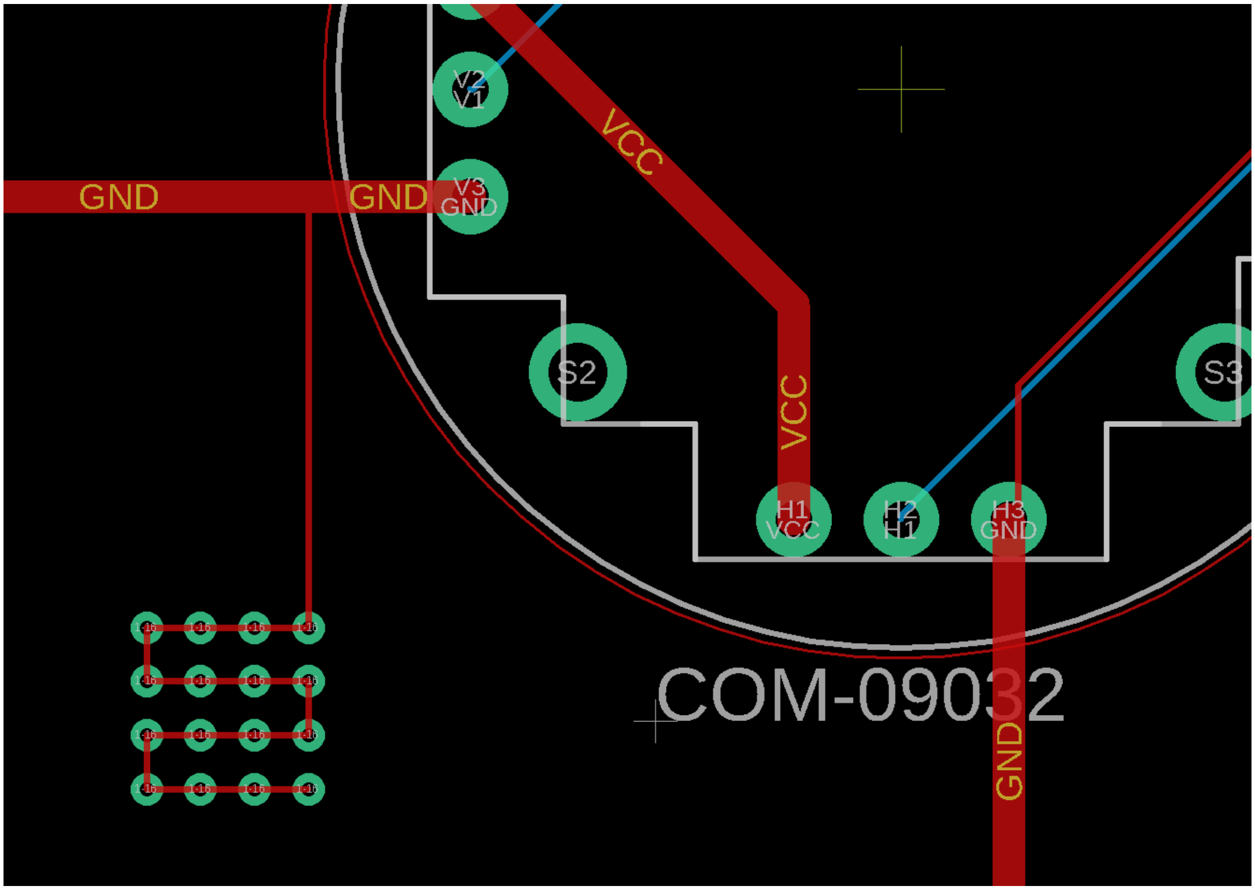

Figure 33 Vias Stitching

8.7 PCB Overall Design

After putting everything together from the last step now we have our final PCB Design ready!

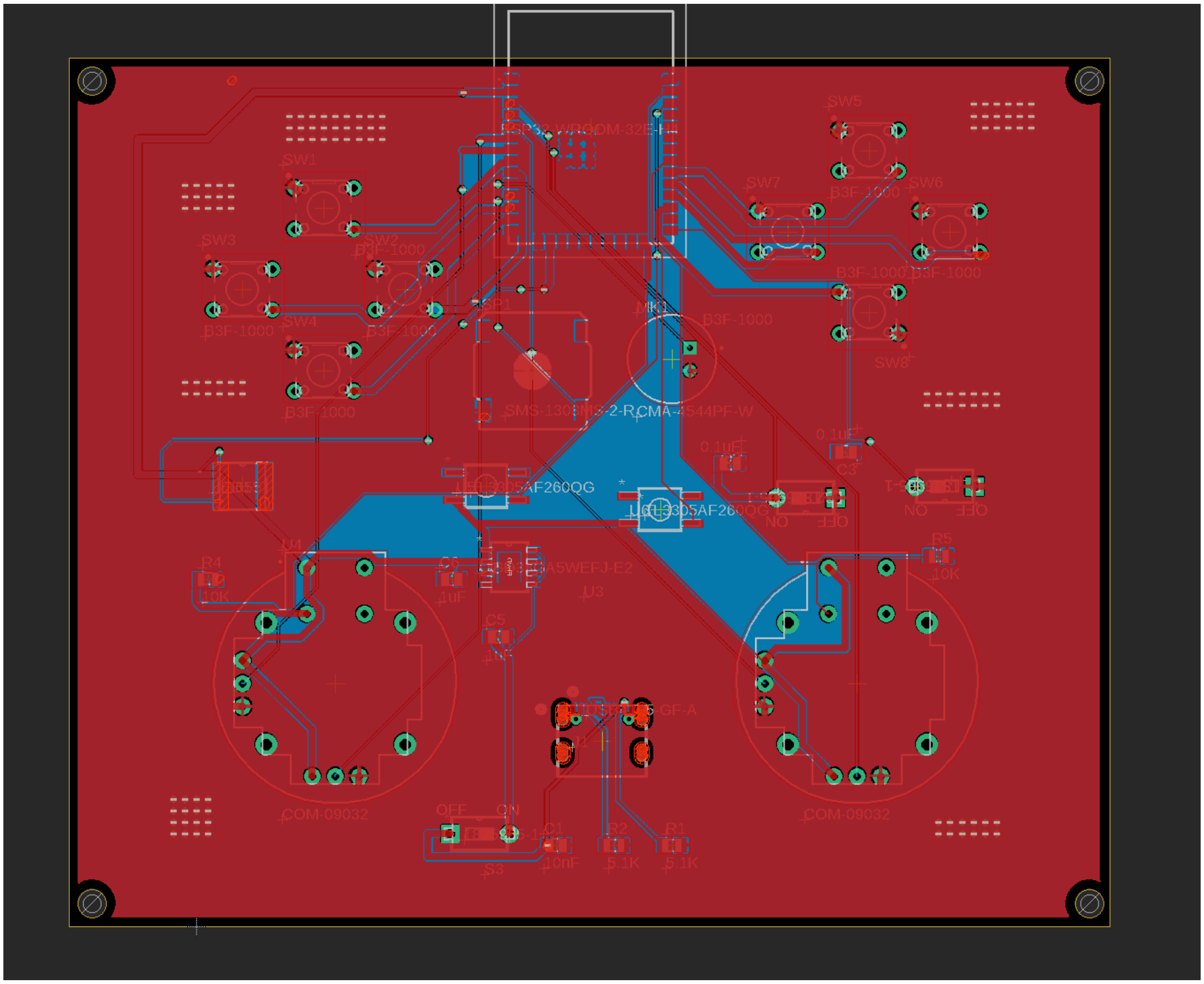

Figure 34 PCB LAYOUT DESIGN

9.0 System Testing

Our testing strategy, divided into distinct phases, meticulously evaluates individual components. Within this structured plan, we ensure that each component's hardware and software aspects undergo rigorous examination. This comprehensive testing plan details our approach to verify and validate every facet of the project, including the assessment of each component in isolation to ensure their individual functionalities perform as intended.

9.1 Hardware Testing

The efficacy of hardware plays a pivotal role in the functioning of any system. Hardware testing stands as the bedrock of ensuring the reliability, stability, and optimal performance of the physical components within a given technology. This critical phase of testing is dedicated to scrutinizing the intricate details and functionalities of the hardware components, assessing their ability to interface with the software, ensuring resilience under varied conditions, and validating their capacity to deliver consistent performance in alignment with the intended specifications.

9.1.1 Speaker

To assess the audio quality of the speaker component, we implemented a meticulous testing procedure. We initiated this process by configuring a specialized RC low-pass filter circuit, strategically designed to filter out unwanted frequencies and provide a clear representation of the speaker's true audio capabilities. The circuit, intricately crafted to optimize signal processing, formed the foundation for our audio quality evaluations.

Beyond audio quality, the volume output of the speaker component was rigorously evaluated. Leveraging online tools designed for accurate sound level measurements, we determined the speaker's maximum loudness. The results of this assessment are presented below, offering insights into the speaker's capability to deliver immersive gaming experiences, particularly in scenarios where heightened volume is paramount.

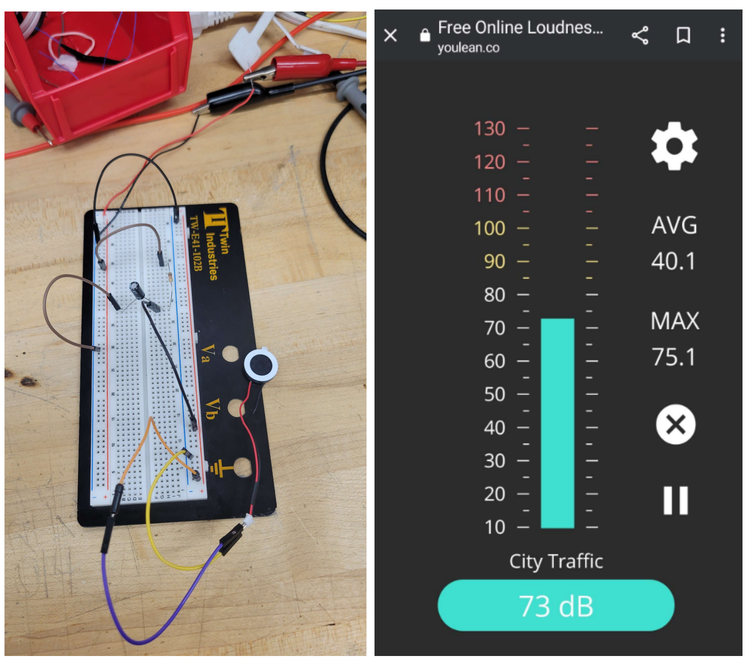

Figure 35 - Speaker Testing

Our assessment of audio quality extended beyond subjective judgments. Precise metrics, including frequency response, which we measure on an oscilloscope. We were trying to provide a quantitative understanding of the speaker's performance. This approach aimed to capture not only the strengths but also any potential limitations in the speaker's audio reproduction capabilities. The image below showcases the graph with milli-Voltage on the Y-axis and milli-seconds on the X-axis.

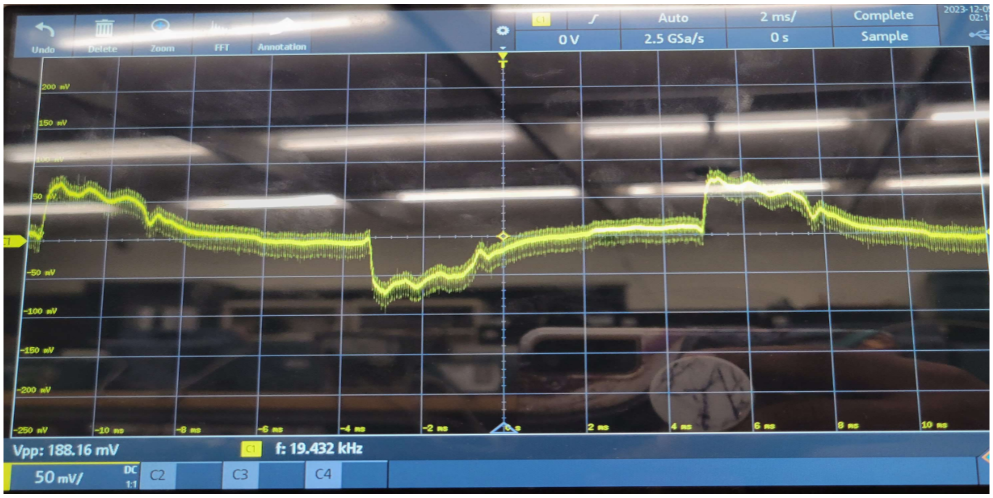

Figure 36 - Oscilloscope Reading

The findings from our hardware tests collectively contribute to a comprehensive analysis of the selected speaker component. This analysis aims to not only validate the component's adherence to a specified performance metrics but also provides valuable insights for potential optimizations or considerations in the overall design of our gaming controller.

9.1.2 Microphone

In this segment, we present the outcomes of the meticulous tests conducted on the microphone component integral to our gaming controller. While audio quality testing was not directly feasible, our focus shifted to a critical parameter—internal resistance. This comprehensive assessment aids in understanding the inherent characteristics of the microphone, shedding light on its electrical properties. Testing the microphone involved a precise measurement of its internal resistance, a key factor influencing its electrical behavior. The objective was to determine how well the microphone aligns with the typical range for electric condenser microphones, which falls between 600**Ω** and 1800**Ω**. To measure the internal resistance, we employed a reliable multimeter capable of accurately gauging electrical properties. The multimeter was strategically connected across the microphone component, creating a circuit that facilitated the measurement of its internal resistance.The recorded results of the internal resistance measurement are presented below, providing a clear insight into the electrical characteristics of the microphone:

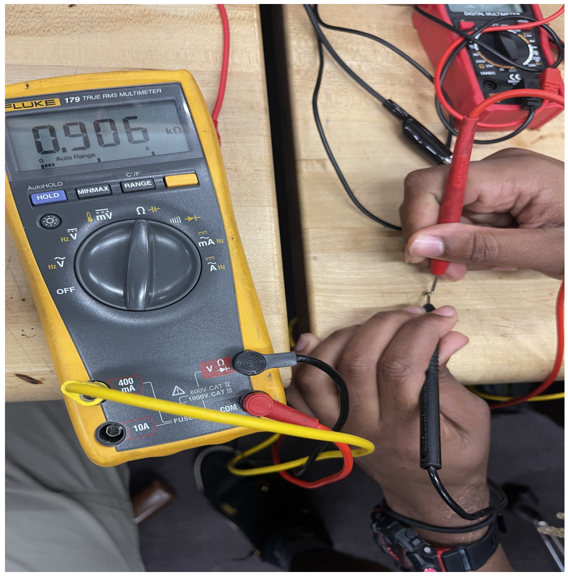

Figure 37 - Multimeter Reading

The audio quality assessment was not directly feasible in this context, the measurement of internal resistance serves as a foundational evaluation of the microphone's electrical behavior. An internal resistance within the expected range suggests compatibility with standard electrical configurations, laying the groundwork for efficient integration into the gaming controller circuit. While internal resistance provides a fundamental understanding of the microphone's electrical properties, future testing endeavors may explore additional parameters such as sensitivity and frequency response for a more comprehensive evaluation of its performance in the gaming controller setup.

9.1.3 LEDs

This section delves into the testing procedures and results pertaining to the LED strip, a crucial element contributing to the visual appeal and functionality of our gaming controller. The focus of this test was to assess the LED strip's responsiveness to power sources and its integration with the action button, unraveling its operational characteristics.

To conduct a comprehensive evaluation, we devised a setup that involved connecting the LED strip to a breadboard. This configuration allowed for controlled power input and facilitated the integration of our action button into the circuit. The action button, in conjunction with the breadboard setup, served as a toggle mechanism to activate and deactivate the LEDs systematically.

The execution of the test involved systematically toggling the LEDs on and off through the action button. The breadboard served as a pivotal intermediary, ensuring the proper distribution of power to the LED strip based on the action button's state. The outcomes of the LED strip test are detailed below, providing insights into its responsiveness to the configured circuit:

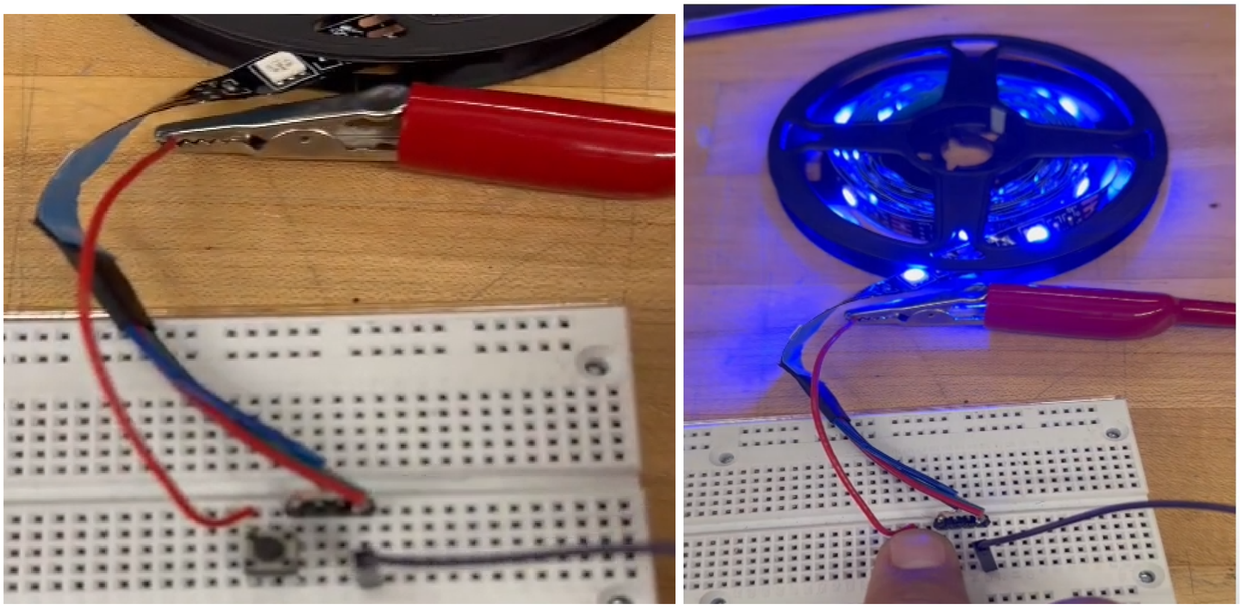

Figure 38 - LED Testing

9.1.4 Power Bank

This section outlines the testing methodologies employed and the subsequent findings pertaining to the power bank component, a pivotal element ensuring uninterrupted gameplay and sustained power for our gaming controller. The test focused on evaluating the power bank's charging mechanism and its response to the completion of the circuit using a push button.

The test setup involved integrating the power bank into a breadboard configuration. This allowed for precise control over the power supply and facilitated the incorporation of a push button within the circuit. The push button served as a trigger to complete the circuit, enabling the flow of power and initiating the charging process for the power bank.

The execution of the test revolved around the systemic activation of the push button to complete the circuit and assess the power bank's response. Simultaneously, the input voltage from the power supply was monitored to gauge the charging process and the power bank's interaction with the circuit. The test outcomes are summarized below, providing insights into the power bank's performance during the charging cycle:

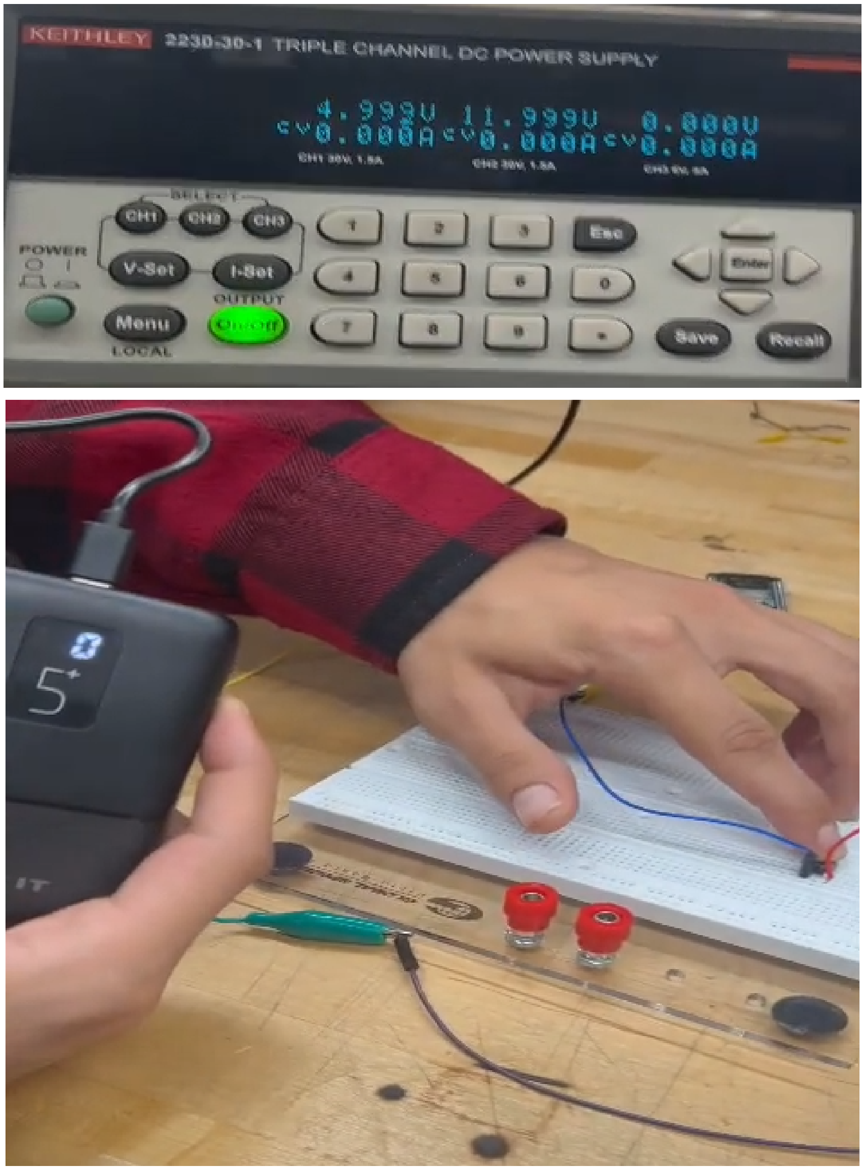

Figure 39 - Power Bank Testing

The analysis of the test results delves into the efficiency of the power bank's charging mechanism, the responsiveness to the push button, and considerations for optimal integration into the gaming controller setup. Considerations for seamlessly integrating the power bank into the gaming controller's circuit are explored, including compatibility with other components and the overall power management system. While the power bank performed successfully we are still weighing our options with a battery pack.

Future testing endeavors may include assessing charging times, power efficiency, and exploring dynamic control mechanisms for the power bank. Recommendations for optimizing the power bank's integration within the gaming controller circuit could stem from observations made during this initial test

9.2 Software Testing

The quality and reliability of each component play a pivotal role in shaping the overall user experience and operational effectiveness of the controller. Rigorous and comprehensive testing serves as the cornerstone of this process, meticulously inspecting, evaluating, and validating the controller's performance, functionality, and endurance.

This critical phase entails a thorough examination of code logic, ensuring it adheres to established coding standards and accurately translates user input. Button responsiveness is meticulously evaluated ensuring consistent and accurate input registration. Wireless connectivity is rigorously tested, ensuring stable connections under diverse usage scenarios. Platform compatibility is verified across various gaming platforms, including PCs, consoles, and mobile devices, ensuring proper button mapping and seamless gameplay across different systems.

Through this comprehensive testing process, potential weaknesses or limitations are identified and addressed, ensuring that the Bluetooth controller delivers an exceptional and seamless gaming experience for users.

9.2.1 Connection Establishment

To thoroughly evaluate the ESP32's Bluetooth connectivity, we test the data exchange between an ESP32 and a PC. This testing process will not only assess the ESP32's ability to establish and maintain a stable Bluetooth connection but also delve into its data transfer capabilities.

The Testing Objectives:

1. Connection Establishment: Verify the ESP32's ability to establish a seamless Bluetooth connection with the PC.

2. Data Transmission: Evaluate the ESP32's efficiency in sending and receiving data packets to and from the PC.

3. Data Integrity: Ensure that data transmitted between the ESP32 and the PC remains intact and free from corruption.

In order to conduct thorough testing of the Bluetooth connectivity, the initial step involves acquiring the Bluetooth Serial Terminal application, available through NMinion on the Microsoft Store. This specific software serves as a crucial tool enabling the evaluation of Bluetooth connections. It provides a user-friendly interface that allows us to establish, monitor, and interact with Bluetooth devices, facilitating the examination of data exchange and ensuring the reliability and functionality of the Bluetooth connectivity. This application acts as a bridge, empowering us to simulate and validate various communication scenarios between devices, aiding in the comprehensive assessment of the Bluetooth functionality being tested.

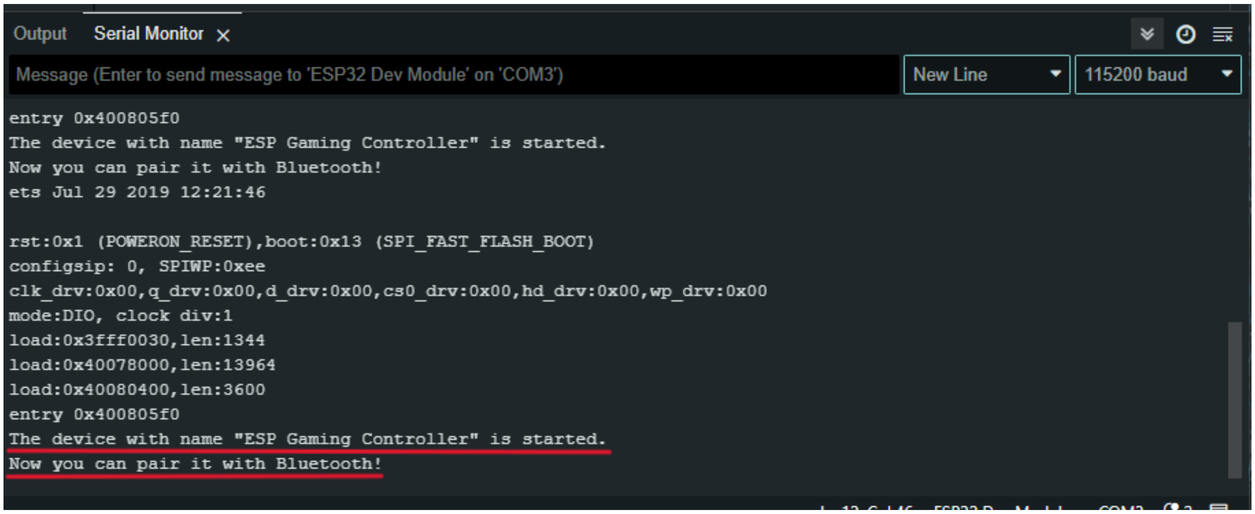

Figure 40 - Bluetooth Discoverable

First we must flash the code to the ESP32 and once that is done the device will be able to be connected to.

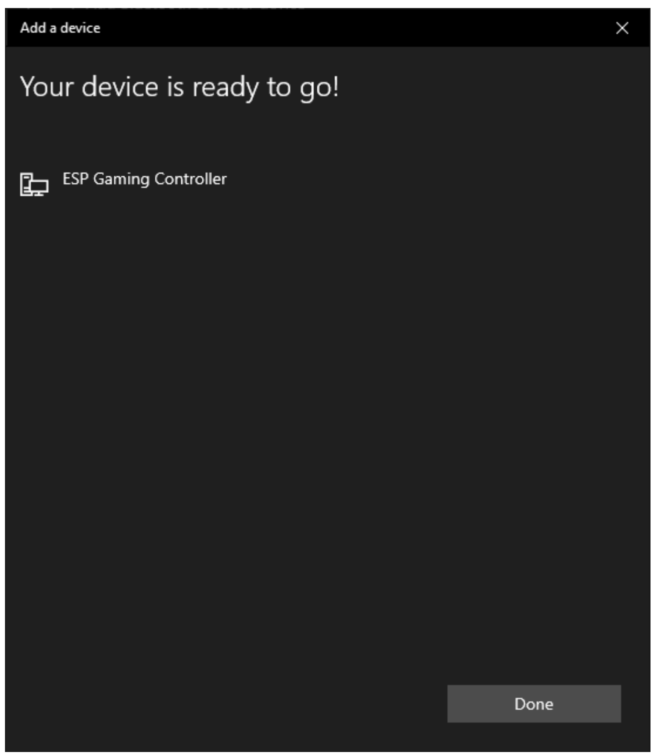

Figure 41 - PC Paired

We then proceed to initiate a connection with the PC. The connection completed roughly around 5 seconds.

Figure 42 - Message from PC

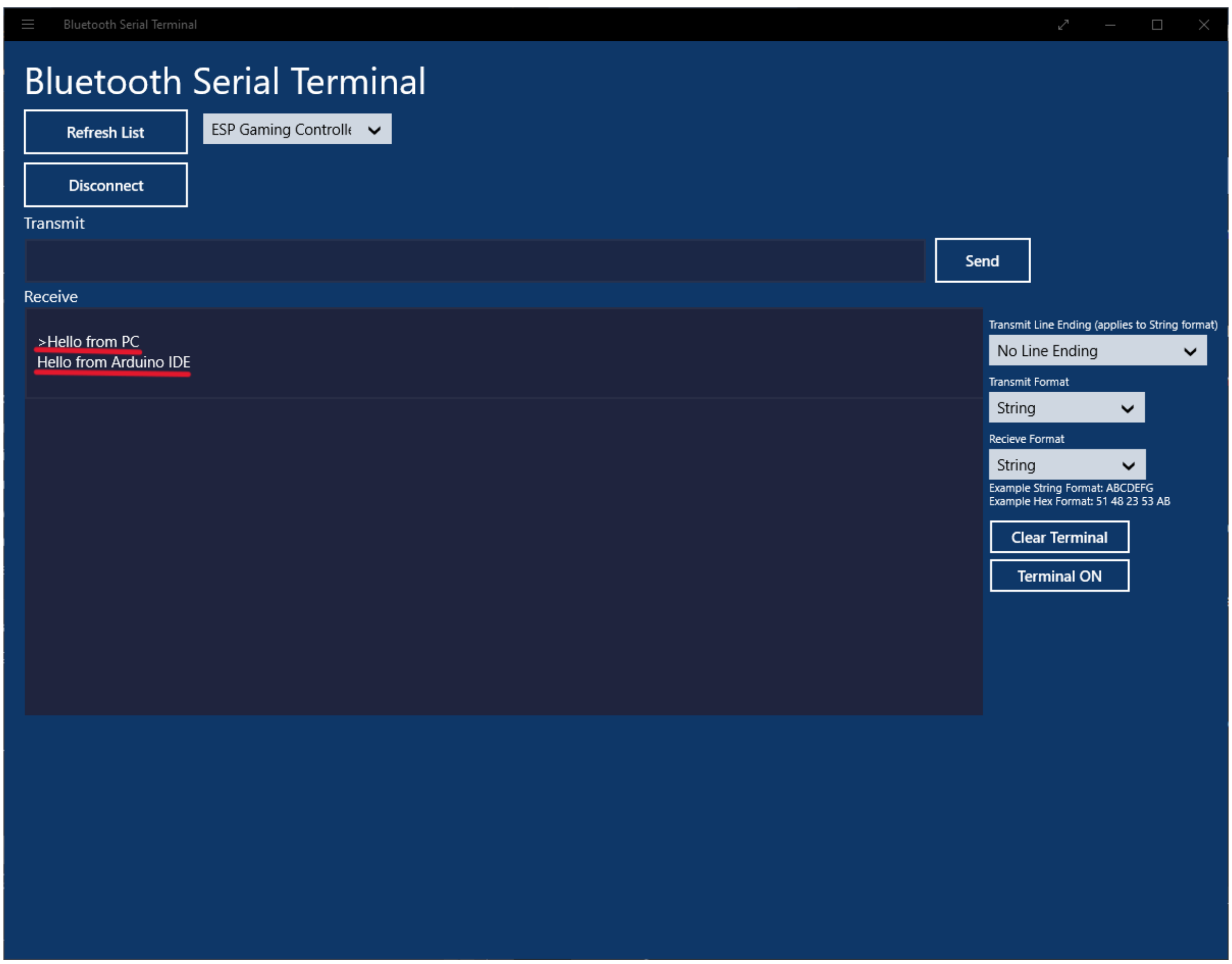

Figure 43 - Message from Arduino

After the Esp32 has made a connection to the PC we begin communicating. On both the ESP32's serial terminal and the PC's own, we should be able to write and receive messages.

The expected outcomes have been successfully realized, affirming the capabilities of the ESP32's Bluetooth functionality. The ESP32 established a robust and dependable Bluetooth connection with the PC within 5 seconds, showcasing its reliability in practical applications. Its ability to efficiently transmit data was evident throughout the testing process, managing various data packet sizes with minimal latency and maintaining data integrity consistently..

9.2.2 Console Switching

This test specifically assesses console switching functionality. During the initial boot-up of the system, it's designed to detect the state of a particular button being held down, triggering a corresponding message output. This action represents the simulation of console switching by identifying a specific button press upon system startup. The code configures the microcontroller to recognize this signal, allowing it to generate a distinct message via the output, indicative of the successful identification of this button's press during the system's power-up phase.

The Testing Objectives:

1. Button Detection Accuracy: The ESP32 should correctly detect the pressed button and trigger the corresponding message upon startup.

2. Button Presses After Boot: The button presses after the ESP32 has been booted should not register and do anything.

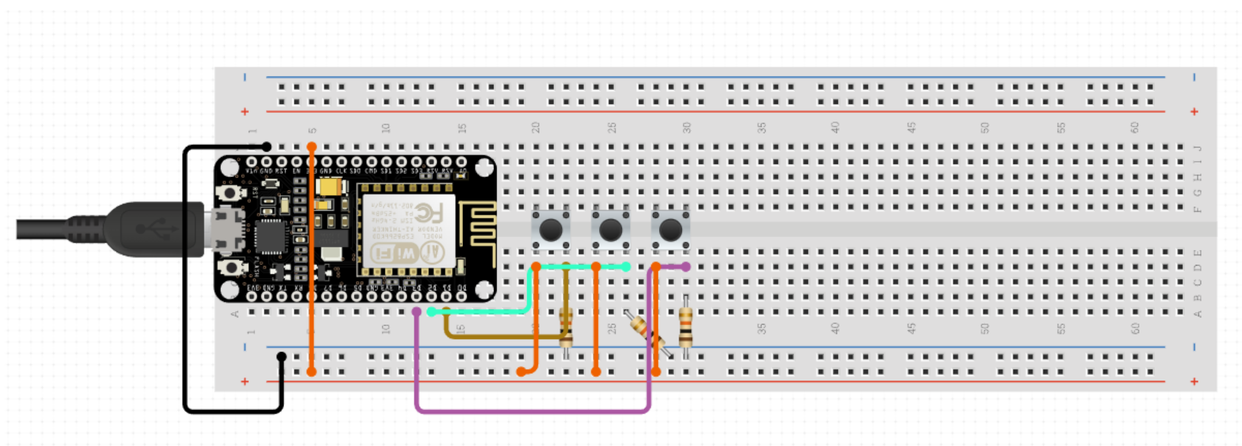

Figure 44 - Console Switch Circuit

In setting up a breadboard configuration with three buttons, each button is connected to an ESP32 microcontroller. The buttons, fundamental input devices, are wired to the breadboard using jumper wires. One leg of each button connects to the ground (GND) rail, ensuring the buttons share a common ground. The other legs of the buttons are each linked to separate GPIO pins on the ESP32, facilitating the detection of button presses. This setup enables the ESP32 to detect when any of the three buttons are pressed or held down. Each button press triggers a specific action or response programmed into the ESP32, allowing for versatile interactions or distinct functionalities based on the button pressed during operation. The arrangement promotes a controlled and responsive user input mechanism, granting the microcontroller the ability to interpret and act upon the signals generated by these three buttons according to the programmed logic.

The program we made begins by defining constants (buttonPin1, buttonPin2, buttonPin3) that represent the GPIO pins connected to three buttons on the board. In the setup() function, the serial communication is initiated, and the pins connected to the buttons are configured as input pins with internal pull-up resistors enabled. Following a brief delay for system stability after booting, the code checks the state of each button. If any button is pressed during startup, determined by a LOW signal due to the pull-up configuration, a corresponding message is sent through the Serial Monitor: "Nintendo Switch" for the first button, "PC" for the second button, or "Android" for the third button.

Holding down “Nintendo Switch Button”:

Figure 45 - Holding down “Nintendo Switch Button”

Holding down “Android Button”:

Figure 46 - Holding down “Android Button”

Holding down “PC”:

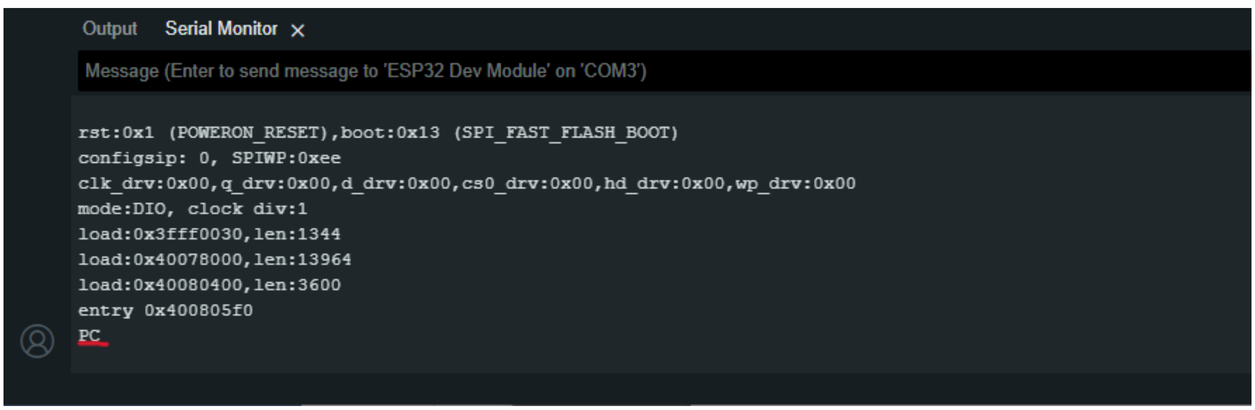

Figure 47 - Holding down “PC”

The code demonstrated commendable accuracy in identifying button presses, consistently executing the anticipated actions upon startup. As intended, the ESP32's functionality to recognize button presses solely during the initial boot sequence was confirmed. Following the boot process, subsequent button presses were effectively disregarded, aligning precisely with our predetermined expectations. This confirmed behavior assured that the ESP32 solely responded to button inputs during its startup phase and remained unresponsive to subsequent presses, meeting our intended design parameters.

9.2.3 Deadzone Configuration

This experiment is aimed at configuring the deadzone of a joystick connected to an ESP32 via a breadboard. The deadzone adjustment helps mitigate small variations in the joystick's resting position to prevent unintended input.

The Testing Objectives:

1. Deadzone Calibration: Assess and fine-tune the deadzone settings on the joystick to determine the optimal range that filters out negligible movements around the center position while responding accurately to intentional joystick actions.

2. Input Accuracy Enhancement: Evaluate the impact of deadzone adjustments on the accuracy and precision of the joystick's input signals. Measure how well the system ignores minor variations in the resting position while effectively capturing intentional movements.

3. Noise Reduction: Investigate how implementing a deadzone minimizes noise or unintended signals generated by the joystick due to factors such as electrical interference or mechanical instability. Measure the effectiveness of the deadzone in mitigating these unwanted inputs.

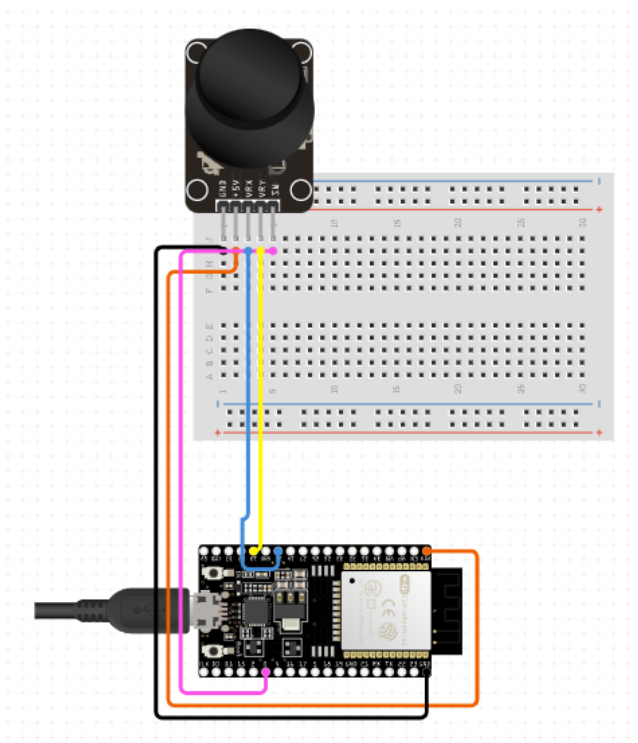

Figure 48 - Deadzone Circuit

We assembled a circuit by connecting a joystick to a breadboard and an ESP32 microcontroller. The joystick's X and Y axes were linked to analog pins on the ESP32 to facilitate analog readings of its movement.

The accompanying code is designed to manage the joystick's input. Firstly, it establishes a deadzone—a neutral area—around the center position of the joystick. Any slight movements within this region are disregarded, preventing inadvertent signals from affecting the output. The program continuously reads the analog values from the joystick's X and Y axes. If these values remain within the deadzone, the system treats the joystick as if it were at rest and centered. This ensures that minute variations around the center do not trigger any action.

Deadzone triggered for X-axis:

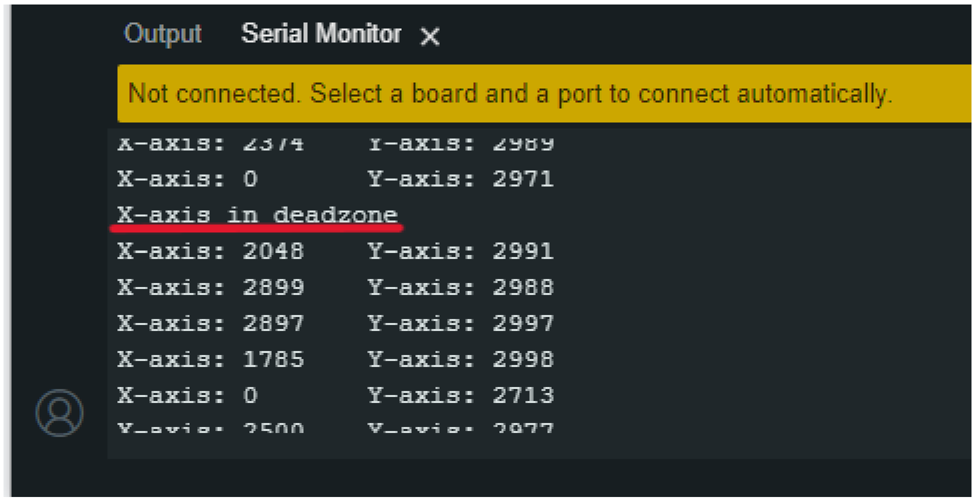

Figure 49 - Deadzone triggered for X-axis

Deadzone triggered for Y-axis:

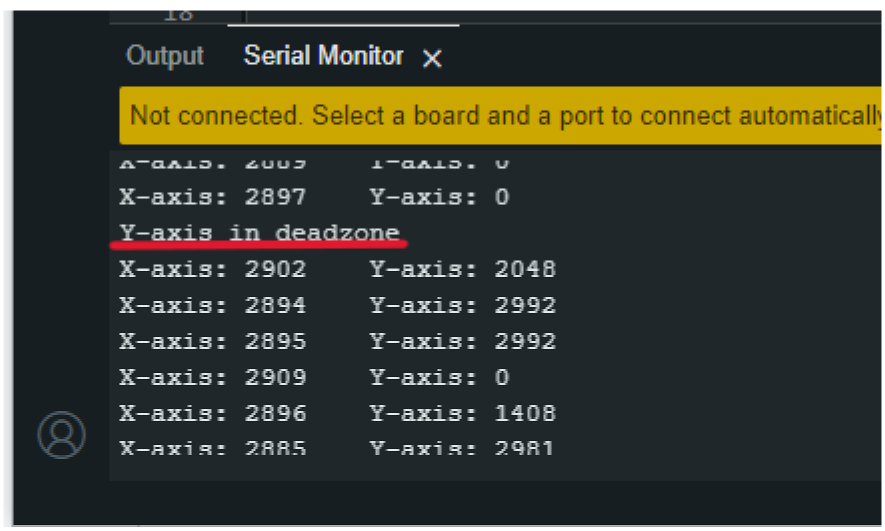

Figure 50 - Deadzone triggered for Y-axis

The experiment was partially successful in its technical execution. However, due to limitations posed by the size of the available breadboard, accessing the pins of only one side of the ESP32 restricted our ability to accommodate the joystick effectively. This constraint altered the center position values, shifting them to approximately 2900 instead of the expected 2048, consequently affecting the entire range of readings and skewing the accuracy of the results.

Despite these limitations, the experiment managed to detect and acknowledge the presence of the dead zone around the altered center values. However, the observed deviations in the input accuracy and noise reduction aspects suggest that the configuration was impacted by the skewed center values, resulting in unexpected behaviors and inaccuracies in discerning intentional versus incidental joystick movements.

The inherent constraints introduced by the small breadboard not only influenced the expected readings but also compromised the precise calibration and evaluation of the deadzone's effectiveness. Consequently, this limitation hindered a comprehensive assessment of the experiment's objectives related to Input Accuracy and Noise Reduction.

In summary, while the experiment demonstrated an awareness of the dead zone, the skewed center values caused by the limited breadboard size significantly impacted the accuracy and effectiveness of the dead zone configuration, affecting the broader assessment of input accuracy and noise reduction aspects.

9.2.4 LED Behavior

This test aims to assess the behavior of the controller's LED system. It's designed to demonstrate how the LEDs respond to battery statuses, distinguishing between charging and non-charging scenarios.

The Testing Objectives:

1. Battery Status Visualization:
    - Assess the LED's ability to reflect and communicate the battery status accurately.
    - Evaluate the effectiveness of LED color representation in indicating different battery life levels (high, medium, low).

2. Charging State Differentiation:
    - Determine if the LED blinking pattern effectively communicates the charging status of the controller.
    - Assess the clarity and recognizability of the LED blinking pattern for users to easily distinguish between charging and non-charging states.

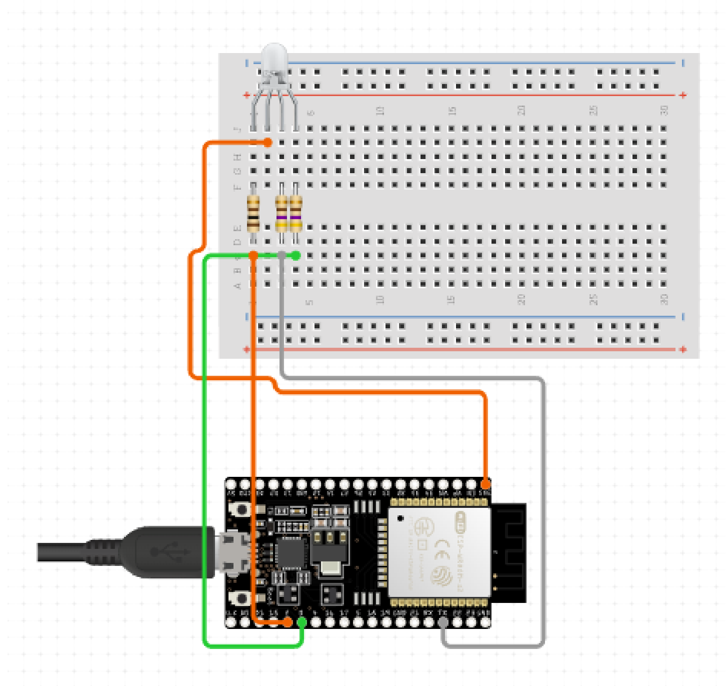

Figure 51 - LED Circuit

We initiated the setup by integrating an RGB LED, strategically paired with 220-ohm resistors, all meticulously connected to a breadboard alongside an ESP32 microcontroller.

The programmed behavior operates as follows: Upon entering "charging" via the input console, the LED swiftly blinks blue for a moment, transitioning back to its default purple hue, repeating this cycle every 5 seconds. Alternatively, typing "not charging" prompts a query for the "Battery Level." Subsequent inputs of "high," "medium," or "low" dictate the LED's blinking behavior—each corresponding to a specific color (green for "high," yellow for "medium," and red for "low"). Similar to the "charging" input, these color-indicating blinks occur momentarily before reverting to the base purple shade, repeating the sequence indefinitely until new input commands are entered.

This setup mirrors a gamepad's visual representation through LED reactions, offering distinct color-coded cues in response to different inputs, thereby providing intuitive feedback to the user. The perpetual blinking cycle remains active until new instructions are introduced through the console interface.

When the “charging” is inputted the LED will blink blue swiftly and return back to purple indefinitely:

Figure 52 - Changing State to Charging

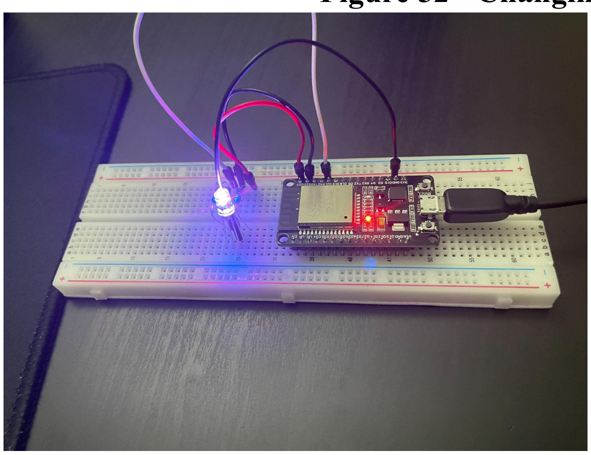

Figure 53 - Base Purple Hue

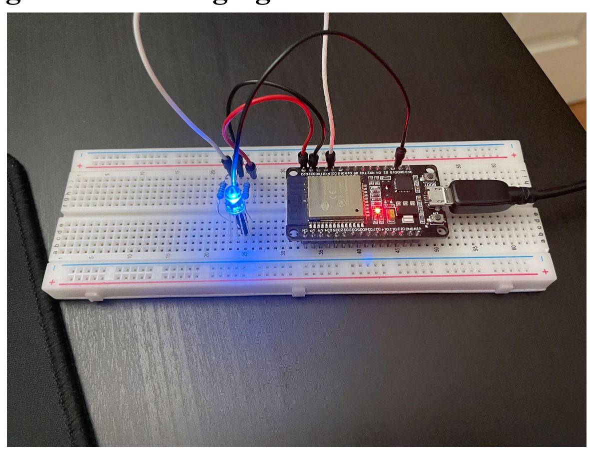

Figure 54 - Blinking Blue

When the "not charging" is inputted the user is prompted "Battery Level." When the user inputs "high", "medium", or "low" the LED will blink green, yellow, or red swiftly and return back to purple indefinitely:

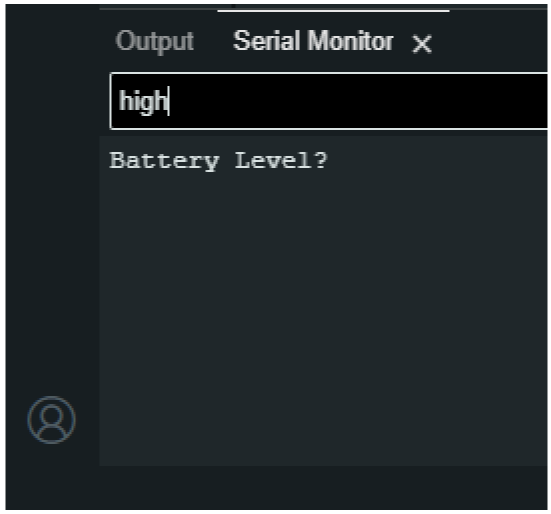

Figure 55 - Changing Battery State to High

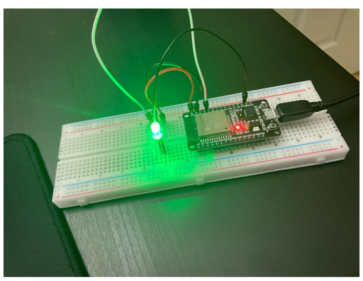

Figure 56 - Green Hue

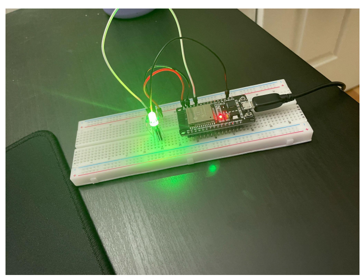

Figure 57 - Yellow Hue

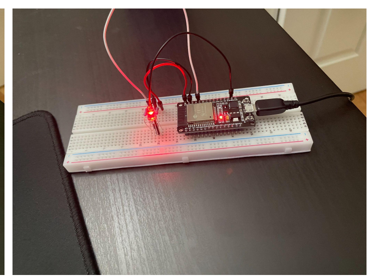

Figure 58 - Red Hue

The experiment yielded successful LED responses consistent with the programmed behaviors. Each user input triggered the intended LED reactions, accurately reflecting a gamepad-like visual representation. The LED system provided clear and intuitive cues in response to different commands, effectively differentiating between charging states, battery levels, and console profiles.

9.2.5 Input Behavior

This test aims to demonstrate the effectiveness of button debouncing by showcasing a scenario where a button press is registered only once despite any bouncing or erratic behavior caused by mechanical switches.

The Testing Objectives:

1. Debouncing Demonstration: The test aims to illustrate how button debouncing works to eliminate unintended multiple state changes when a button is pressed or released due to mechanical bouncing.

2. Single Press Detection: It focuses on detecting a button press as a single event even if the physical button undergoes bouncing during the press or release.

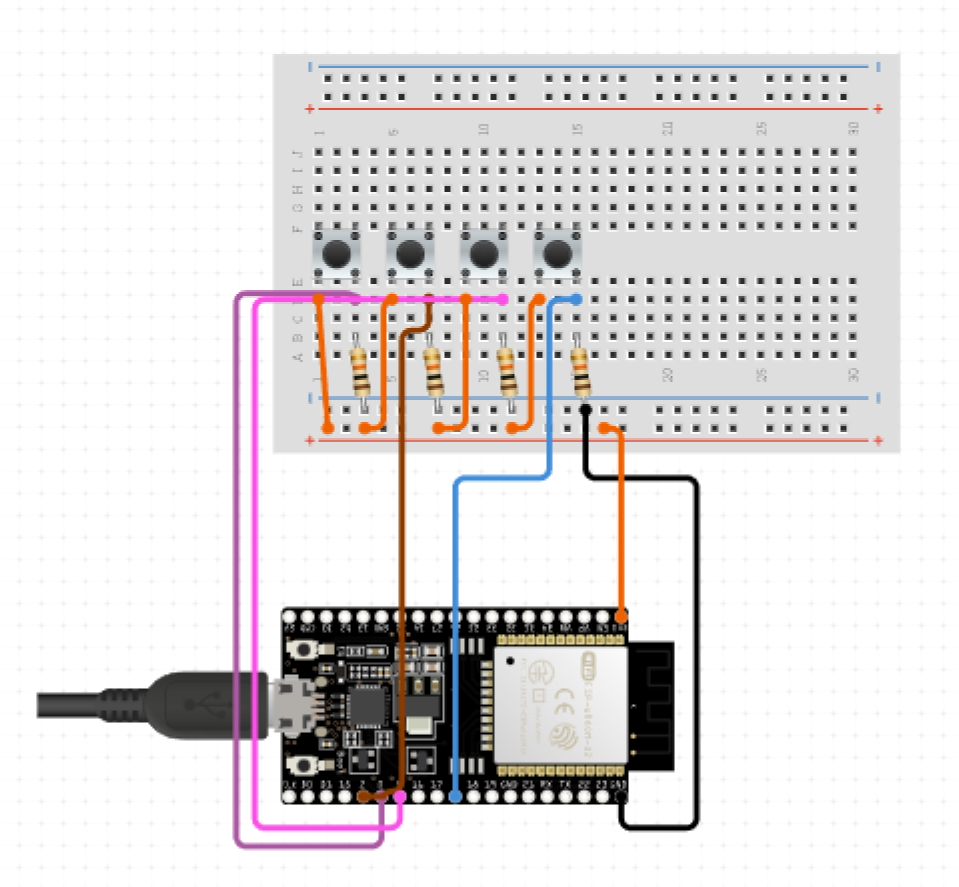

Figure 59 - Button Circuit

In this setup, each push button was connected to the ESP32 via the breadboard. The code running on the ESP32 was designed to detect and manage button presses while mitigating any erratic behavior caused by the mechanical bouncing of the buttons.

The purpose of this setup was to demonstrate how the debounce logic worked in practice. By introducing delays after detecting a change in button state, the code aimed to ensure that a single button press was registered as a coherent event, disregarding any transient fluctuations caused by the physical properties of the buttons.

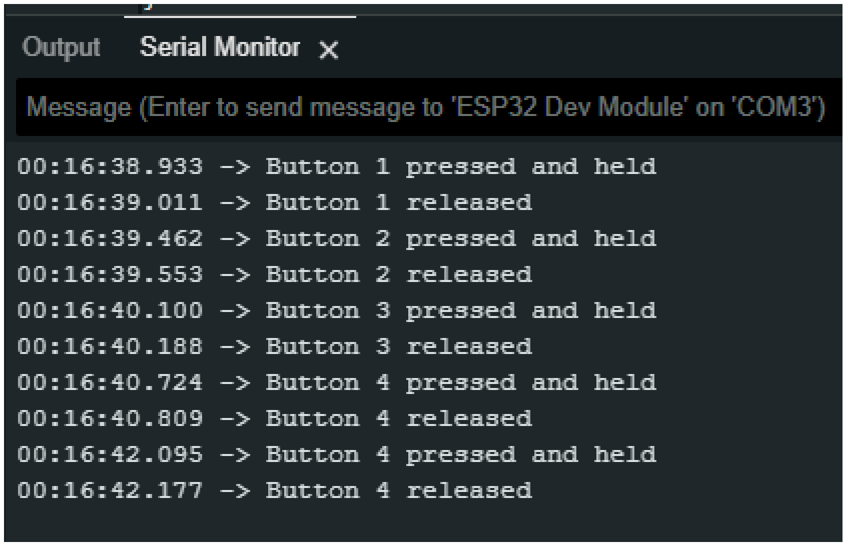

Figure 60 - Debouncing

The test, essentially, verified that despite the potential bouncing effect when a button was pressed or released, the program accurately recognized and responded to a button press as a single, cohesive action. This was evident in the serial output, where the "pressed and held" message appeared only once per button press, and a subsequent "released" message was displayed upon releasing the button.

9.3 Overall Integration

The Multi-Purpose Device Controller system seamlessly integrates various hardware and software components to deliver a reliable and user-friendly gaming experience. The core hardware component, the ESP32 Bluetooth module, ensures stable connections with external devices, efficient data transmission, and data integrity. Button configurations and joystick calibration contribute to accurate user inputs, even though the testing reveals challenges in calibrating the joystick's deadzone due to hardware constraints. The LED system visually communicates battery status and charging states, responding to console inputs and enhancing user awareness.

On the software side, Bluetooth communication logic governs the ESP32, establishing robust connections with external devices. Button and joystick logic translate user inputs into meaningful actions, ensuring the controller responds accurately and consistently. LED control logic interprets console inputs, adjusting LED behavior to communicate real-time information effectively. The implementation of a debouncing algorithm mitigates mechanical bouncing in button presses, guaranteeing accurate detection of single button presses and eliminating unintended state changes.

9.4 Plan for SD2

In Senior Design 2, our main focus is transitioning from the conceptual phase to the practical implementation of our multi-purpose device controller. Building upon the groundwork established in Senior Design 1, our objectives include ordering PCB, soldering necessary components, and creating a fully functional prototype capable of connecting and controlling multiple devices. We will start by finalizing the PCB design by looking at different layout optimizations. Following this, we plan to select a reputable PCB manufacturer, order more components, and distribute soldering tasks among team members based on skill and component. The subsequent phases involve prototyping, testing, and validating the device's functionality, followed by iterative design enhancements based on testing outcomes. Software tests will be conducted on each device to ensure that the MPDC is able to respond correctly to each command for different devices. Comprehensive documentation will be maintained throughout the process, detailing design changes, test results, and encountered challenges. This plan aims not only to achieve the technical goals but also to facilitate a holistic learning experience for team members in the practical aspects of engineering.

10.0 Administrative Content

In this segment, we will delve into the fundamental elements that shape the course of a project, ensuring its success and on-time delivery. From budget estimates to the bill of materials, milestones, and the meticulous organization of work distribution tables, these components constitute the essential framework for project planning and execution. This section will provide a comprehensive understanding of how these administrative components are meticulously prepared and expertly managed to guide projects towards their goals, emphasizing cost-effectiveness, resource allocation, and the timely achievement of milestones. As we explore budgeting, materials planning, project markers, and work distribution strategies, we will uncover the critical strategies that drive efficient project administration, paving the way for project success.

10.1 Estimated Budget

Given the nature of this project, there are multiple components we would have to purchase in order for everything to come together.

Quantity	Value	Package	Order code	Manufacturer	Manuf. Code	Availability	Price (from)
1	10nF	C0805	01J3314	KEMET	C0805C103K5RACTU	1761606	0.032
3	0.1uF	C0805	02AH1357	KEMET	C0805C104K1RECTU	2500	0.064
2	1uF	C0805	03AC2519	MURATA	GRM033R60J105ME	135000	0.005
1	4.7uF	C0805	01N5605	KEMET	C0805C475K3PACTU	54660	0.098
1	ESP32-WROOM-32E-H4	ESP32WROOM3	1965-ESP32-S3-W	DIGIKEY	ESP32-S3-WROOM-1	4983	2.68
1	USB4105-GF-A	USB4105_GCT	56AJ0228	GCT (GLOBAL CONNECTOR TECHNOLOGY	USB4105-GF-A-060	1135	0.398
1	CMA-4544PF-W	CMA-4544PF-W	102-1721-ND	DIGIKEY	CMA-4544PF-W	4028	0.78
2	5.1K	R0805	30AK7375	YAGEO	RT0603BRD075K1L	15370	0.072
2	5.1K	R0805	30AK7375	YAGEO	RT0603BRD075K1L	15370	0.072
3	10K	R0805	01P2111	MULTICOMP PRO	MCTF0805DTC1002	1896	0.006
1	22.1K	R0805	24W7417	TE CONNECTIVITY	RP73PF2A22K1BTDF	870	0.285
1	47.5K	R0805	01AH0258	HOLSWORTHY - TE CONNECTIVITY	RN73C2A47K5BTD	3150	0.415
1	2K	R0805	28K0541	SUSUMU	RR0816P-202-B-T5	6630	0.037
3	1825965-1	SWDIP2_256X19	30AC3538	TE CONNECTIVITY	1825965-1.	98445	0.109
1	SMS-1308MS-2-R	SMS-1308MS-2-	668-1259-2-ND	DIGIKEY	SMS-1308MS-2-R	5494	6.76
8	B3F-1000	B3F-1000	07WX8714	OMRON ELECTRONIC COMPONENTS	B3F-1000	4975	0.206
1	2N706	TO18	06R3470	ONSEMI	MMBTA06LT3G	8649	0.025
1		TO220	13M5705	DIODES INC.	ZXTC2045E6	3024	0.331
2	TPB1,27	B1,27	29AH8519	PANASONIC	10TPB220ML	2500	0.723
1	BD33GA5WEFJ-E2	HTSOP-J8_ROM	52AJ4444	ROHM	BD33GA3WNUX-TR	2837	0.36
2	COM-09032	XDCR_COM-090	1568-1526-ND	DIGIKEY	COM-09032	543	4.5
2	TL3305AF260QG	SMD_F260QG_E	84Y2775	E-SWITCH	TL3305AF260QG	3413	0.076
1	CP2102N-A01-GQFN28	QFN28_5X5_SIL	99AC2808	SILICON LABS	CP2102N-A02-GQFN	27899	2.16

Figure 61 - BOM

These are a general range of prices for specific components on an initial search, given more time cheaper yet functional alternatives can be found. While each component in itself is fairly cheap, as things start to come together, everything can add up quickly, and that's not even to mention any replacement parts we may need to get.

10.2 Initial Project Milestones

Project Specification and Planning (Week 1-3): The first step in our senior design project is to define the project's specifications, objectives, and constraints. We need to finalize the detailed requirements for the gaming controller, including BLE integration, analog joysticks, trigger buttons, batteries, USB-C port, speakers, mic, on/off switch, boost converter, D-pad buttons, and more. We will also establish a clear budget for the project, taking into account component costs.

Schematic Design (Week 9-10): Using Autodesk Eagle or a similar PCB design program, we will create a detailed schematic for the gaming controller. This schematic will serve as the blueprint for the hardware components and their interconnections.

Code Development (Week 10-12): Concurrently, we will start developing the code required for the controller. This includes writing code for joystick calibration, button functionality, and Bluetooth connectivity. Ensuring seamless integration with various devices is a critical aspect of this phase.

Prototyping (Week 13-16): With the schematic and initial code in place, we will begin prototyping the controller. This involves building a basic prototype to test the functionality of key components, such as the analog joysticks and buttons. Early testing will help us identify and address any issues promptly.

Project Documentation (Throughout Semester): Continuous documentation of the project's progress, challenges faced, and solutions devised is essential. This documentation will serve as a valuable resource for troubleshooting and future reference.

Semester 2 Milestones

Component Procurement (Week 1-2): At the beginning of the second semester, we will order the necessary components for the gaming controller. Careful consideration of component prices and alternatives will be essential to stay within the budget.

Assembly and Soldering (Week 3-5): As components arrive, we will commence the assembly process. This includes soldering the PCB, integrating the batteries, and connecting various hardware elements. Attention to detail is crucial during this phase to ensure proper connections and functionality.

Joystick Calibration and Testing (Week 6-8): After assembly, we will focus on fine-tuning the joystick calibration and thoroughly testing all controller functions. It's imperative to eliminate any dead zones or inaccuracies in joystick movement.

Bluetooth and USB-C Connectivity (Week 9-10): We will continue refining the code to enhance Bluetooth and USB-C connectivity. Ensuring that the controller can seamlessly connect to a wide range of devices is a top priority.

User Testing and Feedback (Week 11-12): Involve potential users or testers to gather feedback on the gaming controller's performance and user experience. This feedback will inform any necessary adjustments or refinements.

Final Testing and Validation (Week 13-14): In the final weeks of the second semester, we will conduct comprehensive testing to validate the controller's functionality, battery life, and overall performance. This phase will include stress testing to ensure the controller meets its intended usage requirements.

By adhering to these milestones, we aim to complete our senior design project successfully. The gaming controller will not only meet the specified requirements but also provide an enhanced gaming experience, thanks to features such as the Hall effect joystick and versatile connectivity options.

10.3 Table of Work Distribution

Tasks	Primary	Secondary
PCB Design	Mohammed	Jason and John
Component Specification and Ordering	Jason	Mohammed
Analog Stick and Button Calibration	John	Kenny
Bluetooth Connectivity	Kenny	Jason
Microcontroller Programming	Kenny	John
Hardware Assembly	Everyone	N/A
Battery Regulation	Mohammed	Jason

Table 18

In this project, the division of roles between the team members is a key strategy for success. Kenny and John will spearhead the software development efforts, leveraging their expertise to create seamless and responsive drivers for our gaming controller. Their focus on coding, user experience, and system integration will be instrumental in ensuring that the controller's digital components meet the highest standards. On the other side, Mohammed and Jason will be driving the hardware aspect of the project. Their responsibilities include selecting the right components, designing the physical structure, and overseeing the assembly and quality assurance processes. By collaborating closely, our software and hardware teams will work in tandem to create a gaming controller that seamlessly combines cutting-edge technology with an exceptional user experience, meeting the demands of both gaming enthusiasts and casual players alike.

11.0 Project Summary and Conclusion

In summary, our project centers on developing a Multi-Purpose Device Controller (MPDC) designed to connect with various devices, drawing inspiration from the layout of contemporary gaming controllers such as the XBOX controller. Comprising key components like the PCB, input interfaces, power system, and a central MCU, our device aims to provide users with the flexibility to select and control different devices using specific button combinations through Bluetooth connectivity. Notable features include a battery indicator, voice input through a microphone, and sound output via a speaker.

Certainly, this course has proven to be one of our most challenging experiences to date. Nonetheless, each step, from project selection and motivation to research and implementation, has been immensely enjoyable. Our team, consisting of individuals with diverse skills and passions, collaborated to form a cohesive unit. The selection process was driven by a desire to create a project that not only serves a practical purpose but is also enjoyable for users. Our research involved an exhaustive exploration of existing projects, component selection, microcontroller compatibility, power efficiency considerations, and architectural designs. Balancing considerations like USB protocols, economic constraints, budget, scalability, and environmental impact added layers of complexity.

The PCB design phase, a significant challenge given our team composition, was successfully accomplished using the Eagle software and collaborative efforts. Software selection posed its challenges, and after careful consideration, we settled on C++ for its compatibility with our chosen MCU, the ESP32, and the team's collective expertise. Testing involved both hardware and software components, with a thorough approach to ensure the functionality of critical elements like the speaker, microphone, and ESP32. With successful tests, our confidence in the project's progression and functionality grew, marking a significant milestone in our journey

References

[1] W. Greenwald, “The best video game controllers for every platform,” PCMAG, https://www.pcmag.com/picks/the-best-video-game-controllers (accessed Sep. 25, 2023).

[2] “Home - all controller: The World’s first universal gamepad,” ALL Controller, https://all-controller.com/ (accessed Sep. 25, 2023).

Young, J. (2023, July 28). What is a hall effect controller anyway, and do I really need one?. Windows Central.
https://www.windowscentral.com/gaming/what-is-a-hall-effect-controller-anyway-and-do-i
-really-need-one

TS04-66-50-bk-160-smt Cui devices | switches | DigiKey. (n.d.).
https://www.digikey.com/en/products/detail/cui-devices/TS04-66-50-BK-160-SMT/15634
240

Comparing a step down converter vs Voltage Regulator. Free Online PCB CAD Library. (2022, February 18).
https://www.ultralibrarian.com/2020/08/04/comparing-a-step-down-converter-vs-voltage-r
egulator-ulc#:~:text=A%20step%20down%20converter%E2%80%94also,connected%20d
evices%20or%20electrical%20load.

What is USB-C? an explainer. PCMAG. (n.d.).
https://www.pcmag.com/how-to/what-is-usb-c-an-explainer#:~:text=But%20USB%2DC%
20is%20different,holdout%20session%2C%20the%20Apple%20iPhone.

ESP32 series - espressif systems. (n.d.-a).
https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Sam, R. (2023, October 6). Hall effect joysticks - what is it and why does it matter. TheGamingSetup. https://thegamingsetup.com/guides/hall-effect-joysticks

Evaluating the performance of nickel-metal hydride batteries in electric cars. Energy5. (n.d.).
https://energy5.com/evaluating-the-performance-of-nickel-metal-hydride-batteries-in-elect
ric-cars

C&K switches | Electronic Switches & components manufacturer. (n.d.-a).
https://www.ckswitches.com/media/1422/js.pdf

SN74LVC1G66DBVT Texas Instruments | mouser Europe. (n.d.-c). https://eu.mouser.com/ProductDetail/Texas-Instruments/SN74LVC1G66DBVT?qs=dT9u2OTAaVVwPKfrf5zH0w%3D%3D

Datasheet - STM32WB15CC - Multiprotocol Wireless 32-bit MCU arm. (n.d.-b). https://www.st.com/resource/en/datasheet/stm32wb15cc.pdf

STM32WB15CC. STMicroelectronics. (n.d.). https://www.st.com/en/microcontrollers-microprocessors/stm32wb15cc.html

RX23W - 32-bit Microcontrollers and Module with Bluetooth® 5 for IoT Endpoint Devices, System Control and Wireless Communication | Renesas

SP-3605 Soberton Inc. | audio products | DigiKey. DigiKey. (n.d.). https://www.digikey.com/en/products/detail/soberton-inc/SP-3605/6562949

CMEJ-0733-36-L050 3D model. CMEJ-0733-36-L050 | 3D Model | Microphones | Electret Condenser Microphones |. (n.d.). https://www.cuidevices.com/product/resource/3dmodel/cmej-0733-36-l050

Model: CMEJ-0733-36-L050 | description: Electret condenser microphone. (n.d.-d). https://www.cuidevices.com/product/resource/cmej-0733-36-l050.pdf

MMICT3902-00-012. Component Distributors Inc. (n.d.). https://www.cdiweb.com/products/detail/mmict390200012-tdk-invensense/672362/?pid=568&utm_campaign=12602562379&adgroupid=123539444847&utm_content=508725275177&utm_term=&gclid=Cj0KCQjwy4KqBhD0ARIsAEbCt6iXrlFGq7ATxCaOome2B42IqTCa6hkvCjgsebjtHYIaJTRzzNHb18UaAtWKEALw_wcB

Adafruit 5050 RGB led with integrated driver chip - 10 pack. (n.d.-a). https://www.amazon.com/Adafruit-5050-Integrated-Driver-Chip/dp/B00SK8LY9A

74404042100 Würth Elektronik | inductors, coils, chokes | digikey. (n.d.-a). https://www.digikey.com/en/products/detail/w%C3%BCrth-elektronik/74404042100/4865653

3.1.1a “Xbox Wireless Controller: Xbox.” Xbox.Com, www.xbox.com/en-US/accessories/controllers/xbox-wireless-controller. Accessed 22 Sept. 2023.

3.1.1b Young, Jennifer. “GameSir Launches the First Xbox Controller with Hall Effect Sticks.” Windows Central, Windows Central, 18 July 2023, www.windowscentral.com/gaming/gamesir-launches-the-first-xbox-controller-with-hall-effect-sticks#:~:text=The%20Xbox%20Elite%20Controller%20Series,G7%20has%20Hall%20Effect%20sticks.

3.1.2a "8BitDo Ultimate Controller with Charging Dock." 8BitDo, www.8bitdo.com/ultimate-bluetooth-controller/. Accessed 22 Sept. 2023.

3.1.3a "8BitDo USB Wireless Adapter 2." USB RR 2 8BitDo, www.8bitdo.com/usb-wireless-adapter-2/. Accessed 22 Sept. 2023.

3.2.3.3a Franklin, Curt, and Chris Pollette. "How Bluetooth Works." HowStuffWorks, HowStuffWorks, 11 Nov. 2019, electronics.howstuffworks.com/bluetooth.htm.

3.2.1.2a "What Is USB-C? An Explainer." PCMAG, www.pcmag.com/how-to/what-is-usb-c-an-explainer. Accessed 1 Nov. 2023.

3.3.4a Hall Effect, hyperphysics.phy-astr.gsu.edu/hbase/magnetic/Hall.html. Accessed 5 Oct. 2023.

3.2.5a Mafukidze, Harry. "Introduction to Microcontrollers." Circuit Basics, 9 Dec. 2021, www.circuitbasics.com/introduction-to-microcontrolleres/.

4.1.1a "Understanding USB-C and USB Specifications." C2G, www.cablestogo.com/tech/usb-c. Accessed 15 Oct. 2023.

4.1.2a C++ - Standards, www.open-std.org/jtc1/sc22/wg21/docs/standards. Accessed 25 Oct. 2023.

4.1.3a "The EMC Standards List: Common Standards and Their Significance." Cadence, 11 Aug. 2023, resources.system-analysis.cadence.com/blog/msa2021-the-emc-standards-list-common-standards-and-their-significance.

5.0a Pros & Cons of using chatgpt in higher education. Hurix Digital. (2023, October 11). https://www.hurix.com/chat-gpt-pros-and-cons-of-using-chatgpt-in-higher-education/

7.3.1a Pc, O. (2019, September 15). BLE protocol stack - controller. Medium. https://pcng.medium.com/ble-protocol-stack-controller-2d2d5371deec