Multi-Purpose Device Controller

Johnny Blanc, Jason Jean-Louis, Mohammed Talib Hamed Rashid Al Fahdi, Kenny Dao

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

***Abstract* — This project proposes a novel Bluetooth universal controller that tackles the recurring problem of stick drift in controllers. By utilizing hall effect joysticks and an ESP32 microcontroller, the controller offers superior precision, extended durability, and wireless connectivity for a seamless user experience across various platforms. The project emphasizes meticulous development and efficient resource allocation for a user-friendly and reliable controller design.**

***Index Terms* — Bluetooth Low Energy, Dead Zone, Debounce, Hall Effect, Latency, Polling.**

I. INTRODUCTION

The persistence of stick drift in traditional controllers significantly hinders user experience and precision control. This project addresses this challenge head-on by developing a novel Bluetooth low energy multi-purpose controller utilizing innovative hall effect joysticks powered by an ESP32 microcontroller. Our primary focus is enhancing the reliability and longevity of controller inputs. By integrating hall effect joysticks, which leverage magnetic fields for precise positioning, we aim to entirely eradicate stick drift. This technology ensures consistent and accurate user input, elevating the gaming or device-controlling experience across various platforms. The controller design goes beyond core functionality by incorporating LEDs and a speaker as well. This opens doors for creative use of visual and audio feedback. The controller leverages the capabilities of the ESP32 microcontroller, enabling seamless Bluetooth low energy connectivity for versatile compatibility. This implementation facilitates wireless freedom, granting users increased flexibility and movement during gameplay or device operation. Furthermore, we've incorporated software-based accuracy techniques to maximize responsiveness and eliminate unintended inputs. Dead zone adjustments will allow users to customize the minimum joystick movement required for registration, preventing accidental drift at the center position. Additionally, debouncing techniques will filter out any transient electrical signals that might cause jittery or erratic input readings. These combined approaches ensure the controller delivers the most precise and reliable user experience possible. Through the implementation of hall effect joysticks, meticulous software techniques, and versatile features like LEDs and a speaker, this controller elevates the user experience for gamers and device operators alike. With its focus on accuracy, customization, and immersive feedback, this project paves the way for a new era of control in the ever-evolving world of interactive technology.

II. SYSTEM COMPONENTS

This section dives into controller, exploring the individual components that work together seamlessly to deliver a superior user experience. We'll examine each element, from the innovative hall effect joysticks to the powerful microcontroller, explaining their functionalities and how they contribute to the overall effectiveness of the controller.

A. ESP32

At the heart of the controller lies the ESP32, a powerful System-on-Chip (SoC) boasting a dual-core processor for efficient operation. This powerhouse acts as the controller's brain, interpreting signals from the innovative hall effect joysticks and user interface elements (buttons, joysticks). It then translates this raw data into precise digital information, managing core functionalities like Bluetooth Low Energy (BLE) communication for seamless wireless connectivity. Additionally, the ESP32 has the processing muscle to power optional features like LEDs and speakers, creating a truly interactive user experience.

B. Hall Effect Joysticks

These innovative joysticks represent a significant departure from traditional potentiometer-based designs. Instead of relying on physical contact, which can lead to wear and tear over time, hall effect joysticks utilize magnets and sensors. This contactless approach offers several advantages. First, it eliminates friction, minimizing the risk of the dreaded "stick drift" that plagues traditional controllers. Second, it translates to smoother and more precise control for a superior user experience. This translates to long-lasting performance and reliable control, ensuring your in-game actions are accurately reflected on the screen.

C. Action Buttons

The physical interface between user and controller is formed by the user interface elements. These crucial components translate your intuitive actions into digital commands for the game or device. Our design prioritizes a familiar layout, mirroring industry standards for button placement. This ensures a comfortable and natural grip, even during extended gameplay sessions. High-quality, tactile buttons provide satisfying feedback for each press, offering clear confirmation of your actions. This combination of familiar layout, high-quality buttons, and potential expansion allows for intuitive control and customization, adapting to a wide range of user preferences. The ESP32 microcontroller reads user input from these elements, transforming button presses and joystick movements into precise actions within the game or device, ensuring your commands are accurately translated on the screen.

D. Voltage Regulator

This crucial component acts as a silent guardian behind the scenes, safeguarding the controller's delicate internal circuits. The voltage regulator ensures consistent power delivery throughout the controller's operation. It regulates the incoming voltage (whether from a wall outlet, USB port, or battery) to a safe level for the internal circuits, protecting them from fluctuations that could potentially damage the controller.

E. Speaker

The inclusion of integrated speakers expands the controller's functionality beyond basic control. This feature unlocks a heightened level of immersion for both gamers and device operators. These speakers can significantly enhance gameplay by providing aural cues that complement visual feedback. This innovative feature caters to a wider range of users by adding an immersive audio dimension to the controller's capabilities.

F. LED

Strategically placed LEDs can offer more than just visual flair. They can be programmed to provide vital visual feedback during gameplay or applications notifying users of low battery levels. This allows for quick and clear status updates at a glance, without interrupting the user's focus.

G. Battery

For true freedom of movement, the controller can be equipped with a rechargeable battery. This allows for untethered use, perfect for extended gaming sessions or mobile device operation. The inclusion of a rechargeable battery eliminates the need for constant cable connections, providing a more convenient and versatile user experience.

H. USB-C Charging Interface

Our controller integrates a future-proof USB-C charging port. This user-friendly design allows for effortless connection in any orientation, eliminating the frustration of finding the correct insertion position. This versatile port adheres to the latest industry standards, ensuring seamless compatibility with a wide range of devices and a single cable solution for streamlined power management.

III. SYSTEM CONCEPT

The modern controller has become a familiar tool for gamers and device operators. We respect this established design language while introducing innovative features. Our controller retains the core layout and functionality users expect, but with significant enhancements in key areas like longevity (through hall effect joysticks) and versatility (via the ESP32 microcontroller). This evolutionary approach ensures a smooth learning curve while unlocking superior performance.

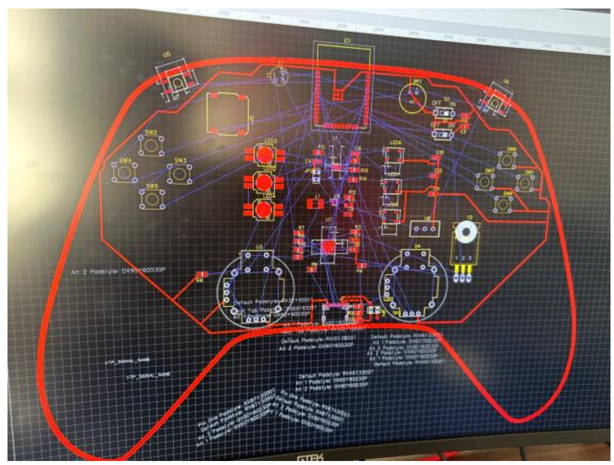


Fig. 1. This diagram showcases the controller's architecture. Buttons, joy sticks, and overall shape resemble the modern gaming controller.

Our controller design draws inspiration from the sleek aesthetics and intuitive layout that have become the hallmark of modern controllers, with a particular nod to the PlayStation 5's DualSense. This translates into a user experience that prioritizes both comfort and familiarity. Meticulous attention has been paid to button placement, mirroring industry standards to ensure a natural and comfortable grip for extended gameplay sessions or device operation. Strategically positioned joysticks further enhance control and responsiveness, adhering to established user expectations.

A powerful ESP32 microcontroller forms the heart of the controller, future-proofing the design. This allows for potential software updates and expanded functionalities as technology continues to evolve. This forward-thinking approach ensures your controller remains at the forefront of user experience, offering exceptional performance for years to come.

Beyond core functionality, we've incorporated strategically placed LEDs that offer more than basic visual feedback. These can be used to indicate battery level, providing you with clear information about your controller's operating status at a glance. Adding a delightful touch to the user experience, the integrated speakers provide a satisfying audio cue when powering on the controller. This subtle audio feedback further enriches the interaction, subtly signaling the controller's readiness.

IV. HARDWARE DETAILS

This section delves into the intricate hardware architecture that underpins the controller's functionality. We'll embark on a detailed exploration of each crucial component, from the CPU housed within the ESP32 microcontroller to the user interface elements that facilitate user interaction. This examination will provide a comprehensive understanding of the hardware's capabilities and how each component synergistically contributes to achieving an exceptional user experience.

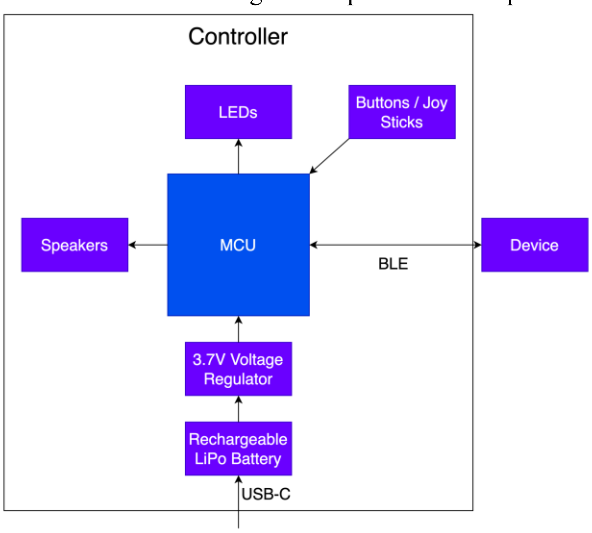


Fig. 2. This diagram shows the controller's core components. The ESP32 chip processes user input from buttons and joysticks, then transmits it wirelessly via Bluetooth. A voltage regulator ensures stable power, and a battery enables wireless use. The controller may also include speakers and LEDs.

A. ESP32 (Microcontroller)

At the heart of the controller lies the ESP32 microcontroller, a powerful and compact System-on-Chip (SoC) acting as the central processing unit. This powerhouse boasts a dual-core 32-bit LX6 CPU with impressive clock speeds reaching up to 240 MHz. This translates to efficient processing capabilities, ensuring smooth operation and rapid response times for even the most demanding gaming or control scenarios. Beyond processing power, the ESP32 integrates Bluetooth Low Energy (BLE) connectivity, facilitating seamless and reliable wireless communication with a vast array of devices. Furthermore, the ESP32 features numerous General-Purpose Input/Output (GPIO) pins. These versatile pins allow the microcontroller to effortlessly manage various peripherals like buttons, joysticks, and LEDs. This combination of processing muscle, wireless connectivity, and peripheral management capabilities paves the way for a truly interactive user experience, seamlessly translating user input into digital signals that power the controller's functionalities.

B. Hall Effect Joysticks

This project incorporates innovative hall effect joysticks, marking a significant departure from traditional potentiometer-based controllers. Unlike potentiometers, which rely on physical contact between a wiper and a resistive track, hall effect joysticks utilize a contactless approach for superior performance. These joysticks employ a magnet embedded within the joystick itself, positioned within close proximity (typically around 1-2 millimeters) to a Hall effect sensor housed within the controller enclosure. As the joystick is manipulated by the user, the magnet disrupts the magnetic field surrounding the sensor. This disruption causes a measurable change in the sensor's output voltage, typically a linear variation. The magnitude and direction of this voltage change correspond to the joystick's position on the X and Y axes. The ESP32 microcontroller, with its 12-bit Analog-to-Digital Converter (ADC), then interprets this analog voltage signal and translates it into a precise digital value. This digital value represents the joystick's position with high resolution, typically ranging from 0 (minimum deflection) to 4095 (maximum deflection) for a 12-bit ADC. This contactless approach eliminates friction and wear-and-tear that plague traditional potentiometer joysticks. Studies have shown that potentiometer-based joysticks can exhibit a phenomenon known as "stick drift" after extended use, where the physical contact between the wiper and resistive track degrades, leading to unintended movement registration. Hall effect joysticks, by eliminating this physical contact, boast a significant reduction in the risk of stick drift, ensuring long-lasting, reliable control with exceptional precision even after extended use.

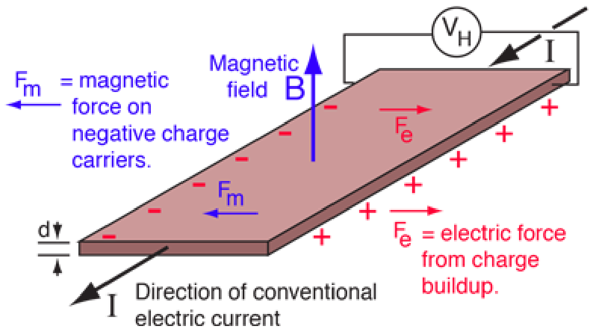


Fig. 3. This diagram shows a hall effect sensor and a magnet. The magnet creates a magnetic field (B) that triggers a voltage (VH) when current (I) flows through the sensor. This principle is used in hall effect joysticks to detect joystick movement for precise control [1].

C. TS04-66-50-BK-160-SMT (Action Buttons)

The controller features high-quality tactile buttons for precise user input. These buttons, model TS04-66-50-BK-160-SMT from CUI Devices, are specifically chosen for their compact size and efficient integration within the controller's design. This 6 millimeter by 6 millimeter footprint allows for a comfortable and familiar button layout that adheres to industry standards. Furthermore, the buttons boast a 5 millimeter actuation height, providing a natural travel distance for button presses. The operating force of 160 grams-force with a tolerance of 50 grams-force offers a balance between requiring a deliberate press for accurate input and minimizing finger fatigue during extended use. Most importantly, these buttons deliver a clear click sensation upon pressing, providing immediate confirmation of user input registration. This combination of compact size, comfortable travel distance, appropriate operating force, and clear tactile feedback ensures precise user input registration while maintaining a familiar and comfortable user experience. The clear tactile feedback further enhances user interaction by providing immediate auditory and physical confirmation of button presses, a critical feature for fast-paced gaming or device control scenarios.

D. WSPH-1805W (Speakers)

The controller integrates WSPH-1805W speakers from Soberton Inc. These speakers are specifically chosen for their compact size and efficient use of space within the controller's design. Measuring only 8 ohms of impedance, they seamlessly integrate with the controller's circuitry. Despite their compact size, these speakers pack a punch, delivering 1W of power. This translates to clear and audible audio feedback, ensuring users don't miss a beat during gameplay or device operation without sacrificing on the controller's portability. Furthermore, the WSPH-1805W speakers boast a frequency range of 300 Hz to 8 kHz. This wide range encompasses the majority of audio cues commonly used in games and device notifications, ensuring clear and informative sound reproduction. By incorporating these speakers, the controller transcends the limitations of visual feedback alone, creating a richer and more engaging user experience.

E. Adafruit 5050 (LEDs)

Our controller boasts a versatile and customizable lighting solution thanks to the incorporation of Adafruit's 5050 RGB LEDs with integrated driver chips. These tiny surface-mount LEDs measure a mere 5mm x 5mm, allowing for easy integration within the controller's design without sacrificing valuable space. Each LED packs a powerful punch, housing three individual light emitting diodes: red, green, and blue. By cleverly combining these three primary colors, a vast spectrum of colors can be displayed, catering to a wide range of user preferences and visual feedback requirements. The integrated driver chip further elevates the user experience. This chip eliminates the need for external resistors, simplifying the overall design and minimizing power consumption for improved efficiency. In essence, the combination of compact size, versatility, and ease of use offered by these LEDs paves the way for creative integration of visual feedback. Imagine illuminating buttons for clear user confirmation, dynamically conveying player health status during gameplay, or even crafting unique lighting effects to enhance immersion. Ultimately, these Adafruit 5050 RGB LEDs with integrated driver chips enrich the user experience by adding a captivating visual dimension to the controller's functionalities.

F. TPS62840DLCR (Voltage Regulator)

Our controller prioritizes consistent performance and component protection through the incorporation of a reliable TPS62840DLCR DC-to-DC high-efficiency voltage regulator module. This crucial component acts as the silent guardian of the controller's internal circuitry, ensuring a steady flow of power throughout operation. Technically speaking, the TPS62840DLCR module functions as a buck converter. This means it effectively steps down a higher input voltage (ranging from 3.2V to 35V) to a lower output voltage (adjustable between 1.25V and 30V), providing a safe and stable power supply for the controller's internal circuits. This adaptability is particularly beneficial, as it allows the controller to operate using a variety of power sources, from USB ports to wall outlets, without compromising performance. The efficient voltage regulation ensures stable operation for all components within the controller, including the ESP32 microcontroller, LEDs, speakers, and any others. Fluctuations in input voltage are a thing of the past, thanks to the TPS62840DLCR module's ability to maintain a consistent and appropriate output voltage. This robust voltage regulator safeguards the controller's delicate components from potential damage caused by voltage spikes or inconsistencies, ultimately promoting reliable and long-lasting performance.

G. LP653042 (Battery)

Our controller incorporates a reliable LP653042 Lithium Polymer (LiPo) battery from EEMB. This rechargeable battery boasts a typical capacity of 820mAh, translating to ample power for prolonged gaming sessions or uninterrupted device operation. The LP653042 is specifically chosen for its compact size and lightweight design, contributing to the controller's overall portability and comfortable handling during extended use. Furthermore, this battery caters to user convenience by supporting various charging methods. A standard micro-USB cable (not included) can be used for recharging, ensuring compatibility with a wide range of readily available power sources. The LP653042 operates at a rated voltage of 3.7V, delivering efficient and consistent power to the controller's components. This ensures smooth operation and an uninterrupted user experience, eliminating worries about sudden power depletion during crucial moments of gameplay or device control.

V. SOFTWARE DETAILS

This section dives deeper into the software written in C that governs the controller's functionality. We'll dissect the software architecture, as visualized in the software diagram, with a specific focus on the algorithms meticulously crafted to optimize performance. We'll explore how these algorithms work to enhance responsiveness and accuracy, ultimately ensuring a seamless and lag-free user experience for the end user.

The software diagram below depicts a flowchart outlining the handshake process between a controller and a compatible device. It utilizes a series of conditional statements to establish and verify a connection. The process begins with the controller initiating the handshake sequence. Following initialization, a conditional branch evaluates whether the device is found. If a compatible device is detected, the handshake proceeds. Conversely, if no device is found, the software loops back to the initialization stage, continuously attempting to establish a connection until successful.

Assuming a device is detected, the handshake phase commences. During this stage, an exchange of data packets occurs to verify compatibility and establish a secure communication channel. Following the data

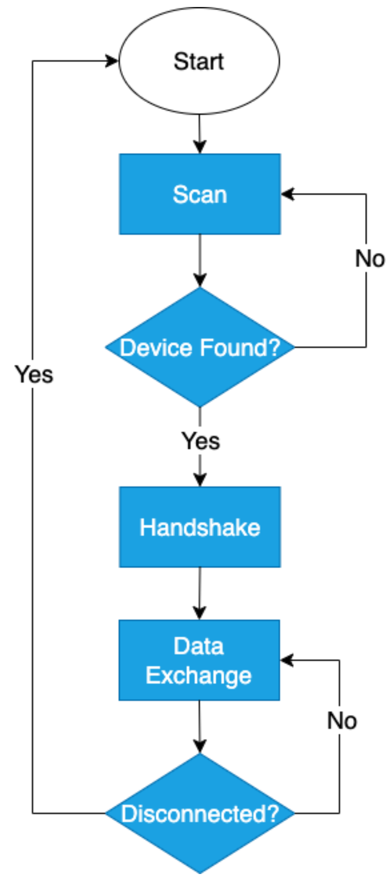


Fig. 4. This flowchart depicts the software handshake process between a controller and a device, outlining the conditional statements, data exchange, and error handling mechanisms that orchestrate a seamless and reliable connection.

exchange, another conditional branch verifies if the handshake is successful. A successful handshake allows for bi-directional communication between the controller and the device. On the other hand, an unsuccessful handshake triggers a loop back to the device detection stage, essentially restarting the connection attempt process.

The software incorporates error handling mechanisms to ensure a robust connection establishment process. In the event of a failed handshake or missing device detection, the software loops back to the appropriate stage to initiate a retry. This iterative approach safeguards against connection failures and ensures the controller persistently attempts to establish communication until successful.

A critical aspect of any controller is its ability to translate user input into precise in-game actions or device controls. In this section, we'll delve into the various techniques employed to minimize errors and maximize the accuracy of user input within our controller design. We'll explore how these techniques address potential sources of inaccuracy, such as signal noise or hardware limitations, ultimately ensuring a responsive and reliable user experience.

A. Polling

Polling serves as the cornerstone of acquiring data from the controller's hardware components, such as buttons and joysticks. Technically, it's a software-driven process where the controller's microcontroller periodically queries these components to ascertain their current state. For buttons, this translates to a binary inquiry – pressed (logical HIGH) or not pressed (logical LOW). Joysticks, on the other hand, necessitate a more nuanced approach. These components typically utilize analog sensors that convert the physical movement of the joystick into a voltage level. The microcontroller's Analog-to-Digital Converter (ADC) then samples this voltage and translates it into a digital value, typically ranging from 0 (minimum deflection) to a maximum value (often 4095 for a 12-bit ADC) corresponding to the joystick's position on the X and Y axes. The polling rate, measured in Hertz (Hz), dictates the frequency of these interrogations. A higher polling rate translates to more frequent updates and potentially lower latency, but also demands greater processing power from the microcontroller. Striking a balance between responsiveness and efficiency is crucial when determining the optimal polling rate for the controller. For instance, a high-performance gaming controller might prioritize a faster polling rate (e.g., 1000 Hz or more) to minimize latency and ensure near-instantaneous response to user input in fast-paced games. Conversely, a controller designed for casual gaming or media control might utilize a lower polling rate (e.g., 125 Hz) to conserve battery life while still delivering an acceptable level of responsiveness.

```
#define BUTTON_PIN  2

uint16_t buttonState = 0;

void readButton(void) {
  (logical HIGH or LOW)
  buttonState = !(PIND & (1 <<
BUTTON_PIN));
}
```

Fig. 5. This C code snippet reads the state of a button (pressed or not pressed) at a specific pin on the microcontroller.

B. Debouncing

Buttons and joysticks, despite their seemingly simple operation, can introduce an unexpected phenomenon known as switch bounce. When a physical button is pressed, the internal contacts may vibrate slightly before fully establishing a stable connection. This transient behavior can cause the electrical signal to register multiple on-and-off events in rapid succession. Imagine a single, deliberate button press registering as a rapid burst of presses and releases – a scenario detrimental to gameplay or device control accuracy. Debouncing techniques meticulously address this issue by incorporating a strategically placed software delay after a button press is initially detected. During this delay, the software deliberately ignores any subsequent state changes from the button. If, after the delay has elapsed, the button press persists (i.e., the electrical signal remains stable), the software confirms a valid press. This approach effectively filters out switch bounce, ensuring the software only recognizes clean and intentional button presses. The debounce time is typically a small value, often measured in milliseconds (ms), and needs to be carefully chosen. A value too low might risk misinterpreting switch bounce as a valid press, while a value too high can introduce a noticeable delay in button registration, hindering responsiveness. Selecting the optimal debounce time involves balancing these factors to achieve a balance between accuracy and responsiveness.

```
  if (currentState != buttonState) {
    lastDebounceTime = millis();
  }

  if ((millis() lastDebounceTime) >=
DEBOUNCE_DELAY) {
    buttonState = currentState;
  }
```

Fig. 6. This C code snippet incorporates debouncing logic. It filters out switch bounce by ignoring rapid button state changes and only updating the button state after a short delay.

C. Deadzone

Joysticks, particularly those employing analog sensors, may not perfectly center themselves when released by the user. Minor variations in manufacturing tolerances or electrical noise can cause the sensor to register a slight non-zero output even when the joystick is intended to be centered. Imagine a scenario where a seemingly centered joystick translates into your character slowly drifting to the left in-game – a frustrating experience to say the least. Deadzones are implemented to address this issue. A deadzone is a virtual area around the center position of the joystick axis where no movement is registered by the software. Joystick movements that fall within the deadzone are essentially ignored. This ensures that unintentional minor movements or electrical noise don't translate into erratic in-game actions or device control. The size of the deadzone can often be adjustable through software settings, allowing users to tailor it to their preferences and playing style. For instance, a competitive gamer might prefer a smaller deadzone for maximum precision in first-person shooter games, while a casual gamer might opt for a larger deadzone to minimize the impact of minor thumbstick movements.

```
if (abs(joystickX) < DEADZONE) {
      joystickX = 0;
}

if (abs(joystickY) < DEADZONE) {
      joystickY = 0;
}
```

Fig. 7. This C code snippet implements deadzones for joystick axes. It ignores minor joystick movements (within a defined deadzone radius) to prevent them from registering as in-game actions.

By meticulously implementing these techniques – polling, debouncing, and deadzones – the controller design strives to minimize errors and maximize the accuracy of user input. This translates to a more responsive and reliable user experience, where players and device operators can be confident that their actions are faithfully reflected within the game or application. They can maneuver characters with precision, execute commands without unintended button presses, and experience a seamless and lag-free interaction that enhances their overall enjoyment and control.

VI. PCB DESIGN

Our printed circuit board has 13 separate circuits. These circuits are divided into few branches. First, the power branch that includes USB-C connector, charge circuit, voltage regulator.

A. USB-C Connector

The USB-C connector serves as the primary interface for power input into the device. It provides a versatile and reversible connection, enabling easy and convenient charging. Additionally, USB-C supports higher power delivery capabilities compared to traditional USB connectors, making it ideal for charging devices with higher power requirements.

B. Charge Circuit (MCP73831-2ACI/MC)

To ensure safe and efficient battery charging, the controller boasts a meticulously designed charging circuit. This circuit centers around the MCP73831-2ACI/MC, a specialized chip that acts as a dedicated charge management controller for single-cell lithium-ion (Li-ion) or lithium-polymer (Li-Po) batteries. This chip offers a multi-pronged approach to optimal charging. It employs a two-stage process: constant current (CC) and constant voltage (CV) charging. Initially, the circuit delivers a precisely regulated current, typically around 500mA, to the battery until a pre-defined voltage threshold (around 4.2V for Li-ion/Li-Po cells) is reached. The circuit then seamlessly switches to constant voltage mode, meticulously maintaining the voltage while the battery current tapers off as it reaches full capacity. Furthermore, the MCP73831-2ACI/MC incorporates thermal regulation. An internal temperature sensor constantly monitors the controller's operating temperature. If it exceeds a safety limit, the charging current is automatically reduced or charging is suspended to prevent overheating. Finally, the controller provides visual or software-based feedback on the charging status, such as an LED indicator or real-time status updates within the connected device or application. The MCP73831-2ACI/MC also communicates with the power source, typically via a USB-C connector, to establish a communication protocol and negotiate the appropriate charging voltage and current. This ensures safe and efficient charging regardless of the specific power adapter or USB port used. In essence, this meticulously designed charging circuit, with the MCP73831-2ACI/MC at its core, fosters a safe, efficient, and user-friendly charging experience.

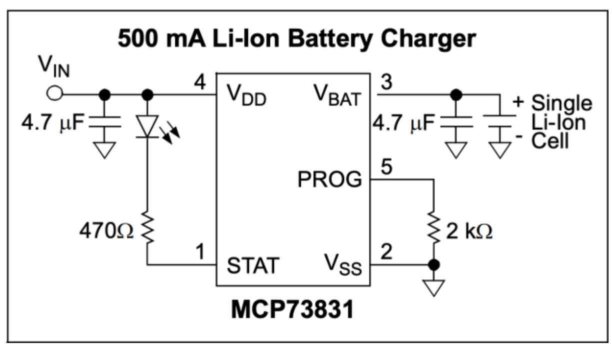


Fig. 8. This flowchart meticulously depicts the software handshake process between a controller and a device, outlining the conditional statements, data exchange, and error handling mechanisms that orchestrate a seamless and reliable connection.

C. Charge Status LEDs

These LEDs provide visual feedback on the charging status of the battery. There are typically three LEDs, each indicating a specific aspect of the charging process:

A. Power Flow LED: Indicates whether power is flowing from the USB-C connector to the charge circuit.
B. Charging Status LED: Indicates the current charging status of the battery, such as charging in progress or standby.
C. Completion of Charge LED: Illuminates when the battery has reached full charge, signaling that the charging process is complete.

D. Voltage Regulator (TPS62840DLCR)

The controller prioritizes stable and reliable power delivery through a meticulously chosen voltage regulator. This vital component, the TPS62840DLCR, acts as a gatekeeper for the incoming power supply. Its primary function is to regulate the voltage supplied to the microcontroller and other downstream components. The TPS62840DLCR ensures the voltage output remains stable and adheres to a specific range that's crucial for proper operation of the MCU. In this instance, the chosen MCU, the ESP32, requires a voltage supply of 3.3 volts. The TPS62840DLCR meticulously adjusts the incoming voltage, regardless of variations in the power source, to consistently provide this critical 3.3 volts. This ensures a reliable and stable power flow that keeps the MCU and other components functioning optimally.

These components work together to regulate the power input, manage the charging process of the battery, and provide a stable power supply to the MCU and other components of the device. The integration of specialized components like the USB4105-GF-A, MCP73831-2ACI/MC, and TPS62840DLCR ensures efficient power management, maximizing the performance and longevity of the device.

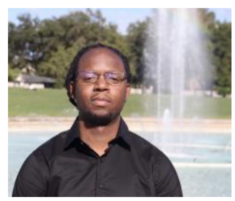
VII. CONCLUSION

This paper has explored the design choices that contribute to a high-performance controller. From the powerful processing core and reliable components to the user-centric features like customizable buttons and immersive feedback mechanisms, each element is meticulously selected to prioritize user experience. The incorporation of Hall Effect joysticks further elevates this commitment to precision by replacing traditional contact-based mechanisms with magnetic sensors. This innovative technology minimizes wear and tear, virtually eliminates drift, and ensures long-lasting responsiveness – critical aspects for competitive gamers and casual users alike.

In essence, this controller design lays a foundation for flawless control. Users can rely on its performance and precision, fostering a sense of immersion and eliminating frustrations caused by lag or misinterpreted inputs. This dedication to both technical excellence and user experience paves the way for a more enjoyable and engaging interaction with games, devices, and applications.

BIOGRAPHY

Jason Jean-Louis Jason is a 25-year-old Computer Engineer from West Palm Beach with hopes of pursuing a career in embedded programming.



Johnny Blanc is a 25-year-old Computer Engineer with hopes of a career in software engineering.

Mohammed Talib Hamed Rashid Al Fahdi is a 24-year-old Electrical Engineer. Mohammed hopes to pursue a career in hardware design and work for major companies that relates to power like Google.

Kenny Dao is a 24-year-old Computer Engineer with hopes of landing a software engineering job. Currently, Kenny is working as an IT service desk intern at a South Florida based company.

REFERENCES

[1] Nave, Carl Rod. "Hall Effect." HyperPhysics, Georgia State University, April 1, 2024. http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/Hall.html