Electrical Overview

Year: 2025 **Semester:** Spring **Team: 1** **Project:**Electronic Skee Ball Machine
Creation Date: February 5, 2025 **Last Modified:** February 5, 2025
Author: Andrew Cali **Email:** acali@purdue.edu

1.0 Electrical Overview

The Electronic Skee Ball Machine utilizes a 32-bit microcontroller (ESP32 Feather V2) to manage system operations, including processing sensor data, controlling actuators, and updating the user interface [1]. Ultrasonic sensors (HC-SR04) detect ball placement and scoring, transmitting data to the microcontroller for real-time processing [2]. The system includes a servo motor (Hiwonder HPS-2018) to adjust the ball launch angle and a DC motor (775 DC Motor), regulated by a Talon SRX motor driver, to control ball launching speed [3][4][5]. An OLED display provides visual feedback [6], while a joystick (D22) allows user input for navigation and gameplay adjustments [8]. Audio feedback is generated through a MAX98357A audio IC to enhance the interactive experience [7]. Data communication is handled through multiple interfaces, with SPI managing display updates, and PWM controlling motor operation [1]. The microcontroller executes software algorithms to process scoring logic, regulate motor behavior, and update the display, ensuring accurate and responsive gameplay.

2.0 Electrical Considerations

2.1 Operating Frequencies

The ESP32 microcontroller operates at 240 MHz, providing ample computational power for real-time control [1]. PWM signals are used for motor control, with the servo motor (HPS-2018) running at 50 Hz and the DC motor (775 DC Motor) controlled at frequencies between 10-20 kHz by the Talon SRX motor driver [3][4][5]. The SPI bus operates at 10 MHz for rapid OLED display updates [2]. The system operates at multiple voltage levels, with the ESP32 and OLED display running on 3.3V [1][6], ultrasonic sensors (HC-SR04) on 5V [2], the servo motor on 6V [5], and the DC motor and motor driver on 12V [3][4].

2.2 Operating Voltage and Power Supplies

The power consumption of each component is calculated to ensure efficient power management. The ESP32 Feather V2 consumes approximately 0.825W [1], while each ultrasonic sensor (HC-SR04) uses 0.075W [2]. The servo motor (HPS-2018) draws 12W, and the Talon SRX motor driver along with the DC motor (775 DC Motor) each require 36W [3][4]. Additional power considerations include the OLED display at 0.0825W [6] and the joystick at 0.033W [8]. Voltage tolerances across the system range from 3.0V to 13V, with appropriate regulation to ensure stability. A 12V power supply (ALT-2410T) is used to support high-power components, while 3.3V and 5V regulators provide stable power for logic components [1][3].

Component	Voltage	Avg. Current Draw	Datasheet Reference
ESP32 Feather V2	3.3V	250mA	0.825W
Ultrasonic Sensor HC-SR04	5V	15mA (each)	0.075W (each)
HPS-2018 Servo Motor	6V	2A	12W
Talon SRX Motor Driver	12V	3A	36W
775 DC Motor	12V	3A	36W
Max98357 Amplifier	3.3V	100mA	0.33W
SOC1602A OLED	3.3V	25mA	0.0825W
Joystick D22	3.3V	10mA	0.033W

3.0 Interface Considerations

Communication between components is facilitated through SPI and PWM interfaces. The SPI bus enables fast data transmission at 10 MHz, ensuring the OLED display updates in real time [6]. PWM signals regulate the servo motor's position at 50 Hz [7] and control the DC motor's speed through the Talon SRX motor driver using a PWM signal at 50 Hz, with a high time between 1-2 ms [3][4]. These interfaces work together to ensure smooth system operation and real-time responsiveness.

Interface	Usage	Data Rate
SPI	OLED display updates	10 Mhz
PWM	Motor control (servo and motor driver)	50 Hz (both)
ADC	Joystick position (potentiometer reading)	N/A (analog signal)
I2C	Speaker control IC	100 kHz

Interface 1: SPI for OLED Display
The ESP32 Feather V2 communicates with the OLED display through an SPI interface. This interface allows the microcontroller to send data to update the display in real time, ensuring the user interface remains responsive. SPI operates at a data rate of 10 MHz, allowing for fast transmission of the display's graphical content and ensuring that the status of the game or system is shown without noticeable delay. The SPI bus uses four signals: Serial Data Out (SDO), Serial Clock (SCK), Chip Select (CS), and Data/Command (DC) to facilitate the communication.

Interface 2: PWM for Motor Control (Servo and DC Motor via Talon SRX)
The PWM interface is used to control both the servo motor and the DC motor. For the servo motor, PWM signals at 50 Hz are used to regulate its position, allowing precise adjustments in angle based on the input signal. The DC motor is controlled through the Talon SRX motor driver using a 50 Hz PWM signal with a high time between 1-2 ms, which determines the motor's speed and power. The Talon SRX interprets the PWM signal from the ESP32 and uses it to control the DC motor accordingly. The use of PWM for both motors ensures efficient and precise control over the system's mechanical components.

Interface 3: ADC for Joystick
The joystick, which functions as a potentiometer, is read using the ESP32's ADC (Analog-to-Digital Converter). The ADC reads the varying voltage from the joystick and converts it into a digital value that the microcontroller can use to determine the joystick's position. Since the joystick operates as an analog input, there is no data rate associated with this interface; the rate at which the ADC samples the joystick will depend on the system's timing requirements and how frequently position data is needed for control.

Interface 4: I2C for Speaker IC
The speaker is controlled through an I2C interface, specifically using the MAX98357 IC. This interface allows the ESP32 to send commands to manage the audio output, such as volume and tone control, as well as triggering sound effects. The I2C bus operates at a clock speed of 100 kHz, providing sufficient bandwidth for communication between the ESP32 and the speaker IC to manage audio functions. This low-speed communication is appropriate given the typical requirements for controlling audio output in your system.

Interface 5: Multiplexing for Ultrasonic Sensors
The ultrasonic sensors are read through a multiplexing technique, where the outputs of the sensors are shared over a common line. Instead of using a communication protocol like I2C or SPI, the ESP32 reads the sensors sequentially by switching the multiplexed lines, which reduces the number of required pins on the microcontroller. This setup doesn't have a defined data rate since the system simply reads the sensor data as it is sequentially switched and measured, making it an efficient method for interfacing multiple sensors with minimal communication overhead.

4.0 Sources Cited:

[1] Adafruit, "Adafruit ESP32 Feather V2 Datasheet," Accessed Feb. 8, 2025. [Online]. Available: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-esp32-feather-v2.pdf

[2] HandsOnTech, "775 DC Motor Datasheet," Accessed Feb. 8, 2025. [Online]. Available: https://www.handsontec.com/dataspecs/motor_fan/775-Motor.pdf

[3] SparkFun Electronics, "HC-SR04 Ultrasonic Sensor Datasheet," Accessed Feb. 8, 2025. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[4] Hiwonder, "HPS-2018 Servo Motor Datasheet," Accessed Feb. 8, 2025. [Online]. Available: https://www.hiwonder.com/products/hps-2018?srsltid=AfmBOorQj4BGvE-_D-zHtcYaxIc6s45n2QYDBvkF3ChVTiyMVc8xEiMt

[5] VEX Robotics, "Talon SRX Motor Controller Datasheet," Accessed Feb. 8, 2025. [Online]. Available: https://www.vexrobotics.com/217-8080.html?srsltid=AfmBOopkhyYHJB0HRQjdzwyNDOoT7HCFYoyhnfY75rNqebHTO3UZaF7G

[6] Technological Arts, "OLED 1602A Datasheet," Accessed Feb. 8, 2025. [Online]. Available: https://www.technologicalarts.ca/support/docs/Components/OLED1602A/SOC1602A.pdf

[7] Analog Devices, "MAX98357A/MAX98357B Datasheet," Accessed Feb. 8, 2025. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/max98357a-max98357b.pdf

[8] Vetco, "VUPN5940 Analog Joystick Module for Arduino D22," Accessed Feb. 8, 2025. [Online]. Available: https://vetco.net/products/vupn5940_analog_joystick_module_for_arduino_d22?srsltid=AfmBOooZZJyYIDsx6wsVB3ZNmTIOSUy-QPGCdJDXSzFrWKKG6CTZjqbB

Appendix 1: System Block Diagram