Software Formalization

Year: 2025 Semester: Spring Team: 1 Project: Electronic Skee Ball Machine
Creation Date: February 17, 2025 Last Modified: February 21, 2025
Author: Maddie Flanagan Email: flanag24@purdue.edu

Assignment Evaluation: See Rubric on Brightspace Assignment

1.0 Utilization of Third Party Software

We will not be using any third party software outside of the STM32 C Standard Libraries

Name	License	Description	Use
STM32F0xx Peripheral Access Layer [1]	BSD 3-Clause License [2]	The STM32F0xx Peripheral Access Layer is a CMSIS-compliant hardware abstraction library provided by STMicroelectronics. It offers direct access to the STM32F0xx microcontroller's registers and peripherals for efficient control.	Used to program and control the STM32F091

2.0 Description of Software Components

All of the following software components have been developed by the team.

Component	Description
Joystick Control	The joystick control system reads analog input from the joystick using the ADC to determine the ball's launch angle. ADC readings range from 0 to 4095 and are linearly mapped to angular displacements between -4° and 4° using a mapping function. A PWM signal is generated using a timer to control the servo motor, with pulse widths mapped between 500ms and 2500ms corresponding to -45° to 45° servo positions. The

	joystick is polled at regular intervals using timer interrupts, ensuring real-time responsiveness to user input.
Button Control	This component detects the duration of a button press to determine launch force. Using GPIO interrupts and a timer, it records the time between button press and release events. A maximum press duration is enforced via timeouts, ensuring consistent game mechanics. If the button is held beyond the maximum allowed time, a timeout is triggered. The resulting press duration is normalized then used for the launch control of the motor.
Launch Control	Integrates the button-controlled launch force with the DC motor. It receives a normalized value between [0, 1] from the button control function, representing the button press duration relative to the maximum allowed time. This value is used to calculate the PWM duty cycle that controls the speed and force of the DC motor responsible for launching the ball. After the button release, the PWM signal drives the motor for a fixed duration before being disabled. This function ensures that the launch force is consistent and responsive to user input while respecting safety and timing constraints.
Display Interface	Interfaces with the NHD-0440AZ 40x4 character display over SPI. Initializes the display by sending specific command sequences to configure modes, clear the display, and set cursor positions. Two sets of SPI commands to control the top and bottom rows. Text is displayed using SPI writes of ASCII characters, with control over cursor positioning and line selection. The display provides the user's current score while playing, and the all-time high score otherwise.
Ultrasonic Sensor Interface	Manages five HC-SR04 ultrasonic sensors for detecting ball positions. The system triggers sensors by sending pulses on their trigger pins and measures the time taken for the echo to return. A timer handles pulse timing and echo duration measurements. A counter iterates through sensors, and timeouts prevent indefinite waits for echoes. The shortest valid echo time identifies which sensor detected the ball, aiding in score determination.
Sound Generation	Uses the I2S interface (SPI2) with DMA to stream audio data from pre-stored WAV files to an external speaker. Audio samples are continuously fed to the I2S interface, with DMA half-transfer and full-transfer interrupts handling real-time buffer refills. Sounds are triggered during specific game events, such as scoring a point.

3.0 Testing Plan

Priority rated from 1 (functionality directly affecting game play) to 6 (non-essential for core function).

Component	Test Plan	Priority
Joystick Control	- Verify ADC readings map accurately to joystick positions (0-4095 → -4° to 4°). - Validate servo movement matches joystick input. - Ensure timer interrupts trigger at correct intervals for real-time responsiveness. - Perform boundary tests at extreme joystick positions such that motor does not rotate past defined boundaries	3
Button Control	- Confirm correct detection of button press, hold, and release events - Test timeout handling for long presses. - Check normalization of press duration to [0, 1] range. - Simulate rapid button presses to assess debouncing.	2
Launch Control	- Test mapping of button press value ([0, 1]) to PWM duty cycle. - Verify motor speed responds correctly to varying duty cycles. - Validate motor stops after fixed duration post-launch. - Perform test integrating joystick and button controls with motor thrust.	1
Display Interface	- Test SPI communication with the NHD-0440AZ display. - Verify initialization sequence correctly configures display. - Check accurate rendering of text, including line positioning and cursor control. - Validate dynamic updates (e.g., score changes) during gameplay. - Test clearing functions and edge cases for text overflow.	5
Ultrasonic Sensor Interface	- Validate correct triggering of ultrasonic sensors. - Check timeout handling for missing echoes. - Test counter for proper cycling through sensors. - Perform detection accuracy tests using varying distances and object positions. - Verify integration with scoring system by updating score onto display.	4

Sound Generation	- Verify SPI2 I2S configuration and DMA setup. - Test playback of different pre-stored WAV files. - Ensure smooth playback without audio glitches. - Test volume consistency and verify accurate audio timing. - Validate correct file parsing from audio file	6

4.0 Sources Cited:

[1] STMicroelectronics, "STM32F0xx Standard Peripheral Library (STSW-STM32048)," *STMicroelectronics*. Accessed: Feb. 22, 2025. [Online]. Available: https://www.st.com/en/embedded-software/stsw-stm32048.html

[2] The Open Source Initiative, "BSD 3-Clause License," *Open Source Initiative*. Accessed: Feb. 22, 2025. [Online]. Available: https://opensource.org/license/BSD-3-Clause

Appendix 1: Software Component Diagram